

**Reference  
Guide**

# hp StorageWorks Storage System Scripting Utility Command View EVA

**Product Version:** 3.2

Fifth Edition (March 2005)

**Part Number:** AA-RU5HD-TE

This guide describes the commands available in the Storage System Scripting Utility (SSSU) for Command View EVA. These commands configure and control HSV controllers.



© Copyright 1998-2005 Hewlett-Packard Development Company, L.P.

Hewlett-Packard Company makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard. The information contained in this document is subject to change without notice.

Compaq Computer Corporation is a wholly-owned subsidiary of Hewlett-Packard Company.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Intel® and Celeron® are U.S. registered trademarks of Intel Corporation.

Intel® and Itanium® are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries and are used under license.

Intel® Itanium™ Processor Family is a trademark in the U.S. and other countries and is used under license.

Microsoft®, MS-DOS®, MS Windows®, Windows®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Hewlett-Packard Company shall not be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Hewlett-Packard Company products are set forth in the express limited warranty statements for such products. Nothing herein should be construed as constituting an additional warranty.

Printed in the U.S.A.

Command View EVA Storage System Scripting Utility Reference Guide  
Fifth Edition (March 2005)  
Part Number: AA-RU5HD-TE

## contents

<b>About this Guide</b>	<b>7</b>
Intended audience	8
Related documentation	9
Conventions	10
Document conventions	10
Text symbols	10
Getting help	12
HP technical support	12
HP storage web site	12
HP authorized reseller	12
<b>1 Introduction</b>	<b>13</b>
Description	14
Formatting and output of returned data	14
Installing the Storage System Scripting Utility	15
Installing the HP-UX kit	15
To execute SSSU	16
Installing the IBM Host Kit	16
LINUX	17
Novell NetWare	17
OVMS	18
SUN	18
TRU64	19
To execute SSSU	20
WINDOWS	20
Starting the Storage System Scripting Utility	21
Syntax	21
Examples	21
Understanding paths and naming conventions	23
Cross Vraid	26

<b>2</b>	<b>SSSU Command Reference</b>	<b>27</b>
	Command introduction	28
	ADD	29
	ADD COPY	30
	ADD DR_GROUP	32
	ADD DISK_GROUP	34
	ADD FOLDER	36
	ADD HOST	37
	ADD LUN	39
	ADD SNAPSHOT	40
	ADD SYSTEM	42
	ADD VDISK	44
	CAPTURE CONFIGURATION	49
	CAPTURE VALIDATE	52
	DELETE	53
	DELETE DR_GROUP	54
	DELETE DISK_GROUP	55
	DELETE FOLDER	56
	DELETE HOST	57
	DELETE LUN	58
	DELETE SYSTEM	59
	DELETE VDISK	60
	EMVERSION	62
	EXERCISE_DISK	63
	EXIT	64
	FILE	65
	HELP	66
	REDISCOVER	67
	RESTART	68
	SELECT	69
	SELECT MANAGER	70
	SELECT SYSTEM	71
	SET	72
	SET DR_GROUP	73
	SET DISK	76
	SET DISK_GROUP	77
	SET FOLDER	79
	SET HOST	80

---

SET MONITOR .....	82
SET OPTIONS .....	83
SET SYSTEM .....	86
SET VDISK .....	87
SHOW .....	90
SHOW DR_GROUP .....	91
SHOW DISK .....	92
SHOW DISK_GROUP .....	93
SHOW HOST .....	94
SHOW LUN .....	95
SHOW MANAGER .....	96
SHOW MONITOR .....	97
SHOW OPTIONS .....	98
SHOW POWER .....	99
SHOW SYSTEM .....	100
SHOW VDISK .....	101
SHOW WORLD_WIDE_NAME .....	102
SHUTDOWN .....	103
 <b>2 Configuration Examples .....</b>	<b>105</b>
Creating a simple configuration .....	106
Creating the storage system .....	106
Creating and presenting a virtual disk .....	108
Using the SHOW commands .....	110
SHOW SYSTEM .....	110
SHOW VDISK .....	111
SHOW HOST .....	112
SHOW LUN .....	113
 <b>A Glossary .....</b>	<b>115</b>
 <b>Index .....</b>	<b>125</b>
 <b>Tables</b>	
1 Document Conventions .....	7





## about this guide

This reference guide describes the HP StorageWorks Storage System Scripting Utility (SSSU) commands.

“About this guide” topics include:

- [Intended audience](#), page 8
- [Related documentation](#), page 9
- [Conventions](#), page 10
- [Getting help](#), page 12

## Intended audience

This guide is intended for storage administrators who are experienced with the following:

- Enterprise Virtual Array v2.006, v3.0, v3.010, v3.014, and v3.020
- Configuration of SAN fabrics
- HP-UX, HP Open VMS, HP Tru64, Microsoft Windows, Sun Solaris, IBM AIX, Novell Netware, and Linux AS and SLES8 operating systems



## Related documentation

This section lists documents that provide additional information:

- *HP StorageWorks Host Operating System Kit for Enterprise Virtual Array Installation and Configuration Guide* (available for Sun Solaris, IBM AIX, Windows, Tru64 UNIX, OpenVMS, HP-UX, Novell Netware, and Linux operating systems)
- *HP StorageWorks Enterprise Virtual Array Users Guide*
- *HP StorageWorks Enterprise Virtual Array - How to Get More Information*
- *HP StorageWorks Enterprise Virtual Array Release Notes*
- *HP StorageWorks Enterprise Virtual Array Storage System Read Me First*
- *HP StorageWorks Enterprise Virtual Array Storage System World Wide Name Label*
- *HP StorageWorks Interactive Help for command view eva*
- *HP StorageWorks Enterprise Virtual Array Configuration Guide*
- *HP StorageWorks Enterprise Virtual Array Upgrade Instructions*
- *HP StorageWorks Enterprise Virtual Array License Instructions*
- *HP StorageWorks Enterprise Virtual Array Catalog of Associated Documentation*
- *HP StorageWorks Enterprise Virtual Array Storage System Installation Instructions*
- *HP StorageWorks Heterogeneous Open SAN Design Reference Guide*

## Conventions

Conventions consist of the following:

- [Document conventions](#)
- [Text symbols](#)

### Document conventions

The document conventions included in [Table 1](#) apply in most cases.

**Table 1: Document conventions**

Element	Convention
Cross-reference links	Blue text: <a href="#">Figure 1</a>
Key and field names, menu items, buttons, and dialog box titles	<b>Bold</b>
File names, application names, and text emphasis	<i>Italics</i>
User input, command and directory names, and system responses (output and messages)	Monospace font COMMAND NAMES are uppercase monospace font unless they are case sensitive
Variables	<monospace, italic font>
Web site addresses	Blue, underlined sans serif font text: <a href="http://www.hp.com">http://www.hp.com</a>

### Text symbols

The following symbols may be found in the text of this guide. They have the following meanings.



**WARNING:** Text set off in this manner indicates that failure to follow directions in the warning could result in bodily harm or death.

---



**Caution:** Text set off in this manner indicates that failure to follow directions could result in damage to equipment or data.

---

**Note:** Text set off in this manner presents commentary, sidelights, or interesting points of information.

---

## Getting help

If you still have a question after reading this guide, contact an HP-authorized service provider or access our web site: <http://www.hp.com>.

## HP technical support

Telephone numbers for worldwide technical support are listed on the following HP web site: <http://www.hp.com/support/>. From this web site, select the country of origin.

---

**Note:** For continuous quality improvement, calls may be recorded or monitored.

---

Be sure to have the following information available before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

## HP storage web site

The HP web site has the latest information on this product as well as the latest drivers. Access storage at: <http://www.hp.com/country/us/eng/prodserv/storage.html>. From this web site, select the appropriate product or solution.

## HP authorized reseller

For the name of your nearest HP-authorized reseller:

- In the United States, call 1-800-345-1518
- In Canada, call 1-800-263-5868
- Elsewhere, see the HP web site for locations and telephone numbers: <http://www.hp.com>.

# Introduction



This chapter introduces the HP StorageWorks Enterprise Storage System Scripting Utility (SSSU) for the HP StorageWorks Enterprise Virtual Array. The following topics are covered:

- [Description](#), page 14
- [Installing the Storage System Scripting Utility](#), page 15
- [Starting the Storage System Scripting Utility](#), page 24
- [Understanding paths and naming conventions](#), page 26
- [Cross Vraid](#), page 29

## Description

The Storage System Scripting Utility (SSSU) for Enterprise Virtual Array is a command line application that allows you to configure and control EVA 3000 and 5000 controllers.

You can execute configuration requests using Command View EVA or SSSU. Command View EVA is the graphical user interface that allows you to control and monitor a storage system. Use Command View EVA to handle simple or initial configuration tasks easily and expediently. Use SSSU to script and run repetitious and complex configuration tasks.

## Formatting and output of returned data

All returned SHOW command data is formatted in the form of:

`XMLtag : Data`

The XML tag displays before the data, and then the tag's data displays.

## Installing the Storage System Scripting Utility

The SSSU is available at the following url:

[http://h20000.www2.hp.com/bizsupport/TechSupport/ProductList.jsp?locale=en\\_US&taskId=135&prodSeriesId=435210&prodTypeId=12169](http://h20000.www2.hp.com/bizsupport/TechSupport/ProductList.jsp?locale=en_US&taskId=135&prodSeriesId=435210&prodTypeId=12169)

To download the files for your operating system from the website:

1. Click either product from **HP OpenView Storage Operations Manager v1.2** list:  
    HP Storage Operations Manager v1.2 Media/Doc Kit  
    or  
    HP Storage Operations Manager v1.2 Migration Media
2. Click **Microsoft Windows 2000** (below **select operating system**).
3. Find your operating system in the revision (rev) column of the Software System Management table and click **obtain software**.

Once installed, SSSU can run as a simple executable. For convenience, you can add the directory containing the SSSU executable to your path or copy the executable to a directory already in your path. Ensure that the SSSU executable's attributes are set with the correct security and execution flags according to your system.

The executable file is named *sssu* or *SSSU.EXE* depending on the operating system.

## Starting the Storage System Scripting Utility

SSSU starts at a command prompt window, shell, or equivalent.

---

**Note:** To run SSSU, you must set up password access to the element manager from the management appliance. You cannot set this password from within SSSU. The [SET OPTIONS](#) command on page 83 controls how SSSU behaves. The options you set with the **SET OPTIONS** command are in effect for the current session only. Each time you start SSSU, the default options are reinstated.

---

---

**Note:** The [CAPTURE CONFIGURATION](#) command on page 49 is the only way to capture, save, and recreate a storage system's configuration. After you have successfully created a storage system, use the **CAPTURE CONFIGURATION** command to create scripts that you can use to recreate a storage system in the event of failure or to create an exact replica.

---

## Syntax

```
SSSU <additional arguments>
```

If SSSU is started without arguments, a generic application `<NoSystemSelected>` prompt is displayed on the terminal and input can be accepted.

If SSSU is started with arguments, those commands are echoed to the input terminal and executed, and then the utility exits.

---

**Note:** You must enclose in double quotes any commands or object names that have embedded blanks (spaces).

---

## Examples

```
SSSU
SSSU "FILE snapD1.txt"
SSSU "cmd1" "cmd2" "cmd2"
```

The first example starts SSSU without additional arguments and prompts you for commands.



The second example starts SSSU and then executes the file *snapD1.txt* from the current directory.

The third example starts SSSU and executes multiple commands.

## Understanding paths and naming conventions

This section describes important information about paths and naming conventions in SSSU.

---

**Note:** For detailed descriptions of the following commands, refer to [Chapter 2](#).

---

The following important root folders let you organize your storage system:

- “\Hosts\”
- “\Virtual Disks\”
- “\Disk Groups\”
- “\Data Replication\”
- “\Hardware\” - This folder exists within SSSU; however, you cannot create any objects within this folder.

You cannot create root-level folders. You can, however, create additional folders within the “\Hosts\” and “\Virtual Disks\” folders to organize your storage system (see “[ADD FOLDER](#)” on page 36).

---

**Note:** SSSU requires that you qualify specified names with full paths. If, however, you are using the default placement when adding Hosts and Virtual Disks, you do not need to include the full path.

---

### Hosts Examples

For example, to add a host named *engineering* to the root Hosts folder:

```
ADD HOST engineering WORLD_WIDE_NAME=1111-2222-3333-4444
```

Whenever you refer to this host, you must give the full path. For example, when adding a LUN to this host:

```
ADD LUN 4 HOST=\Hosts\engineering VDISK="\Virtual  
Disks\Yekao\ACTIVE"
```

If you create a folder structure within the root Hosts, you must include all levels of the folder structure in your commands.

Here is an example of how to add the engineering host to the resources folder within the Host root folder:

```
ADD HOST \Hosts\resources\engineering WORLD=1111-2222-3333-4444
```

### Virtual Disk Example

Because of the space in the root name for the Virtual Disks folder, you must always enclose strings that include the name of this folder within double quotes (“”).

---

**Note:** Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

---

For example, to add a virtual disk family `new_code` to the root folder Virtual Disks:

```
ADD VDISK "\Virtual Disks\new_code" SIZE=10
```

This example creates a 10-GB virtual disk family in the default disk group `new_code`.

---

**Note:** This is a virtual disk family. The actual virtual disk name (the active virtual disk) is `\Virtual Disk\new_code\ACTIVE`.

---

If you create a virtual disk (ADD VDISK) within a deeper folder structure, you need to include this full path within double quotes.

```
ADD VDISK "\Virtual Disks\engineering\gene_research" SIZE=2
```

### Disk Groups Example

When a path includes the root folder “\Disk Groups\”, you must enclose the entire path in double quotes:

```
SHOW DISK_GROUP "\Disk Groups\admin"  
  
ADD VDISK "\Virtual Disks\new_code" SIZE=10 DISK_GROUP="\Disk  
Groups\admin"
```

---

**Note:** Although this guide shows commands spanning more than one line, always enter the SSSU command on one line.

---

**Note on OS\_UNIT\_ID and CONSOLE\_LUN\_ID**

This value is used for IBM AIX (set to zero), OpenVMS (required), and Tru64 UNIX (recommended). Other host operating systems ignore the value. See the host operating system installation guides for more information.

**Note on changing comments on a disk enclosure**

SSSU does not allow comments to be changed on a disk enclosure. Anyone needing to do this can change them using the CommandView EVA UI.

## Cross Vraid

Cross Vraid for Snapshot and Snapclone allow customers to change the Vraid type when creating snapshots or snapclones for better disk utilization when making redundant copies of data. Cross Vraid is supported in the following manner:

- Cross Vraid Snapshot ([ADD SNAPSHOT](#)) and Snapclones ([ADD COPY](#)) within the same disk group.
- Cross Vraid Snapclones ([ADD COPY](#)) across disk groups (including standard and FATA disk groups)

The SSSU v3.2 and Command View EVA v3.2 along with VCS v3.020 are required to support Cross Vraid Snaps.

Refer to [ADD COPY](#) and [ADD SNAPSHOT](#) for more information.



# SSSU Command Reference

## 2

This chapter defines all the commands available in SSSU, including:

- [ADD](#), page 29
- [CAPTURE CONFIGURATION](#), page 49
- [DELETE](#), page 53
- [EMVERSION](#), page 62
- [EXIT](#), page 64
- [FILE](#), page 65
- [HELP](#), page 66
- [RESTART](#), page 68
- [SELECT](#), page 69
- [SET](#), page 72
- [SHOW](#), page 90
- [SHUTDOWN](#), page 103

In addition, this chapter includes syntax and examples. The commands are presented in alphabetical order.

## Command introduction

Most of the commands have switches that you can also use. These are described under the entry for the command in this chapter.

You can display a list of switches for each command directly within the interface by entering a ? after the command or option name. For example, `ADD ?` displays all of the switches available with the `ADD` command, and `ADD SYSTEM ?` displays the switches available for the `ADD SYSTEM` command (see “[Getting help](#)” on page 12).

Remember when you issue commands:

- Use the full path to qualify specified names.
- If a path name contains a space, enclose the entire name in double quotes (“”).

For a full discussion of these requirements, see “[Understanding paths and naming conventions](#)” on page 23.

[Appendix A](#), “[Configuration Examples](#),” provides a simple configuration example for using these commands to create and present a single LUN to a host. It also includes sample output from `SHOW` commands.

---

**Note:** Commands are not case sensitive, and it is often not necessary to type complete command names. For example, you can type `sho` for `SHOW` or `shu` for `SHUTDOWN`.

---



**ADD**

Use the ADD commands to create new systems, folders, disk\_groups, dr\_groups, virtual disks, copies, snapshots, as well as to add LUNs and hosts within the Enterprise Virtual Array.

---

**Note:** Creating copies, dr\_groups, and snapshots of virtual disks is dependent on your licensing level. See the documentation that came with your hardware for more information.

---

## ADD COPY

Use the ADD COPY command to create a copy of the specified virtual disk. A copy is a new virtual disk family. The ADD COPY command is equivalent to creating a snapclone within Command View EVA.

### Syntax

```
ADD COPY <copy_name> VDISK=<vdisk_family>
```

### Switches

The following switches are available with the ADD COPY command.

#### **DISK\_GROUP=**

This is the name of the disk group where you want to create the virtual disk copy. The disk group must already exist to use this switch. If not specified, SSSU uses the same group as the source VDISK.

#### **OS\_UNIT\_ID=**

The ID that is presented to the host operating system. If set to zero, no ID is presented to the host.

#### **REDUNDANCY=**

The amount of data protection set when creating the virtual disk. If not specified, the default is Vraid0.

- **Vraid0**—Provides no data protection. It distributes data among its member disks into stripes and uses all members to process I/O requests. This method has no overhead associated with duplication of information and provides the highest performance.
- **Vraid1**—Provides the highest level of data protection but uses the most space. It duplicates data written to one disk onto a backup disk. In a multidisk configuration, Vraid1 mirrors each pair of disks to each other. These disk pairs can then be striped to create a virtual disk.
- **Vraid5**—Provides a moderate level of data protection. This method distributes the parity information among all disk members. If one drive fails, the failed disk can be recreated after it is replaced.

**VDISK=**

This is the virtual disk name to be copied.

**WORLD\_WIDE\_LUN\_NAME=**

Sets the World Wide LUN Name on unpresented storage.

**WAIT\_FOR\_COMPLETION**

Waits until the command's operation completes before displaying the next SSSU prompt (and returning control) or before running another script line. For large (greater than 1 TB) virtual disks, using this option can result in a long period before the prompt displays.

**NOWAIT\_FOR\_COMPLETION**

Default. Returns control as soon as possible. Some operations are invalid until the initial operation completes in the background.

**Required Switch**

The following switch is required with the ADD COPY command:

**VDISK=**

**Examples**

```
ADD COPY wednesday_nite VDISK="\Virtual Disks\payroll\ACTIVE"  
ADD COPY save_reports VDISK="\Virtual Disks\daily_biz\ACTIVE"  
DISK_GROUP="\Disk Groups\small_disks"
```

The first example copies the virtual disk *payroll* to a copy *wednesday\_nite*.

The second example creates a copy of *daily\_biz* as *save\_reports* within the specified disk group.

---

**Note:** When you use the ADD COPY command, you are actually creating a VDISK. If you want to delete the VDISK, you must use the DELETE VDISK command.

---

## ADD DR\_GROUP

Use the ADD DR\_GROUP command to create a data replication group containing the specified source virtual disk. This source VDisk will be replicated on the specified destination system.

### Syntax

```
ADD DR_GROUP <dr_group_name> VDISK=<vdisk> DESTINATION  
SYSTEM=<system>
```

### Switches

The following switches are available with the ADD DR\_GROUP.

#### **ACCESSMODE=**

The access rights for a connected host. Possible values are readonly and disable. The default is disable, and this is the safest setting.

---

**WARNING:** Setting ACCCESSMODE to readonly could seriously impede host OS operation.

---

#### **COMMENT=**

Used to attach comments to the DR Group being created. This string must be enclosed in quotes. The maximum number of characters for this optional switch is 64.

#### **DESTINATION\_SYSTEM=**

The system on which the destination VDISKS are created on linked systems within this DR Group.

#### **DESTINATION\_DISK\_GROUP=**

The disk group name in which the VDISK on the destination system is created.

#### **DESTINATION\_VDISK\_NAME=**

The name of the VDISK that is created on the destination system. The default name is the same as that of the source.

**VDISK=**

The name of a source system VDISK to be placed within the DR Group. A destination VDISK is automatically created on the destination system.

**WRITEMODE=**

Defines the I/O interaction between the destination and source side. Possible values are synchronous and asynchronous. The default is synchronous.

**Required Switches**

The following switches are required for the ADD DR\_GROUP command.

- DESTINATION\_SYSTEM =
- VDISK =

**Example**

```
ADD DR_GROUP Transactions DESTINATION_SYSTEM=Makawao  
VDISK="/Virtual Disks/Hawaii/ACTIVE"
```

This example creates a DR\_GROUP named *Transactions* that contains the source virtual disk "*Virtual Disks\Hawaii\ACTIVE*." This DR\_GROUP will be connected to the destination system *Makawao*, on which the destination virtual disk "*Virtual Disks\Hawaii\ACTIVE*" (same as the source by default) will be created and added to the destination DR\_GROUP.

## ADD DISK\_GROUP

Use the ADD DISK\_GROUP command to add disk groups to the system.



**Caution:** Vraid1 and Vraid5 are virtual disk families that have a SINGLE or DOUBLE group spare policy to ensure that there is adequate space to regenerate data. If you set the spare policy to NONE, the disk group may not have enough available space to regenerate Vraid1 and Vraid5 data.

---

### Syntax

```
ADD DISK_GROUP <group_name>
```

### Switches

The following switches are available with the ADD DISK\_GROUP command.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed, and the string must be enclosed in double quotes.

#### **DEVICE\_COUNT=**

The number of physical disks to use for this group. The limit is the number of available disks in the storage system. The default and minimum value is 8.

#### **DISKGROUP\_DISKTYPE=**

This parameter determines the types of disks to be considered for creating the disk group. The default value is ONLINE.

ONLINE: Online Fibre channel disks are considered for creating the default disk group.

NEAR-ONLINE: Near-Online Fibre channel disks are considered for creating the default disk group.

**OCCUPANCY\_ALARM=**

The point when a defined percentage of space is used. When this point is reached, an event log is generated and sent to the Management Appliance (optionally the host system) informing the administrator that the group is becoming full. Do not use the percentage sign (%) after the number. The default is 95%.

**SPARE\_POLICY=**

This parameter determines the amount, if any, of storage space set aside for use in the event that disks fail. The default is single.

- **NONE**—Reserves no space within a disk group to allow for data reconstruction in case of failure of disk drives
- **SINGLE**—Reserves space within a disk group to allow for data reconstruction for failure of a single disk drive
- **DOUBLE**—Reserves space within a disk group to allow for data reconstruction for failure of two disk drives

---

**Note:** The space set aside is not in numbers of physical disks. It is the equivalent amount of storage space spread across all disks.

---

**Example**

```
ADD DISK_GROUP "\Disk Groups\human_resources" DEVICE_COUNT=12  
SPARE_POLICY=SINGLE OCCUPANCY_ALARM=75
```

This example creates a new disk group, *human\_resources*. It has 12 physical disks with the equivalent of one disk set aside as a spare and sends an event log to the Management Appliance when 75% of capacity is reached.

## ADD FOLDER

Use the ADD FOLDER command to create a new folder within the specified folder to aid in organizing your storage system. You can create folders only under the “Virtual Disks” and “Hosts” root folders. You cannot create root folders.

For example, if you have a controller that is serving *HR* and *Engineering*, you could create four folders—two to separate the virtual disks and two to separate the hosts:

- “Virtual Disks\Engineering” and “Virtual Disks\HR”
- \Hosts\Engineering and \Hosts\HR

Creating these folders allows you to put engineering virtual disks and hosts in the Engineering folders and Human Resources storage/hosts in the HR folders. This makes it easier to keep track of the pieces within your storage system. If you want to nest folders inside folders, you must add one folder at a time.

### Syntax

```
ADD FOLDER <folder_name>
```

### Switch

Only the COMMENT switch is available with the ADD FOLDER command.

#### COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

### Examples

```
ADD FOLDER \Hosts\human_resources
ADD FOLDER "\Virtual Disks\colorado\colorado
springs\engineering"
```

The first example creates a new folder *human\_resources* within the root folder “Hosts.” The second example creates a folder *engineering* in the folder *colorado springs*, which is two levels below the root folder “Virtual Disks” (“Virtual Disks\colorado\colorado springs”).

The folder structure in the second example must already have been in place before creating the new folder *engineering*.



## ADD HOST

Use the ADD HOST command to add a host and its World Wide Name (WWN) to the list of hosts that can connect to virtual disks within the current system.

The ADD HOST command adds the first Fibre Channel Adapter (FCA) only. The [SET HOST](#) command (see page 80) adds each subsequent FCA.

### Syntax

```
ADD HOST <host_name> WORLD_WIDE_NAME=<world_wide_name>
```

### Switches

The following switches are available with the ADD HOST command.

#### IP=

This is the network IP address. If the IP address is not specified, the name of the host will be specified as is (DNS).

#### COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed, and the string must be enclosed in double quotes.

#### OPERATING\_SYSTEM=

The type of operating system for the specified host. Choose one of the following operating systems (they are spelled here as SSSU expects):

##### ■ CUSTOM

The correct syntax is

```
OPERATING_SYSTEM=CUSTOM=<16 hex characters>
```

##### ■ HPUX

##### ■ IBMAIX

##### ■ OPEN\_VMS

##### ■ SOLARIS

##### ■ TRU64

##### ■ UNKNOWN

## ■ WINDOWS

For the default operating system, refer to Agent Options - User Interface Options in the user interface.

### **WORLD\_WIDE\_NAME=**

The World Wide Name (WWN) of the fibre channel adapter (FCA).

### **Required switch**

The following switch must be set to ensure that Command View EVA recognizes the host added with the ADD HOST command.

WORLD\_WIDE\_NAME

### **Example**

```
ADD HOST \Hosts\development WORLD_WIDE_NAME=5000-1fe1-ff00-0000
```

This example adds a host *development* with an adapter at a WWN of *5000-1fe1-ff00-0000*.

## ADD LUN

The ADD LUN command makes previously created VDISKS available to a host.

### Syntax

```
ADD LUN <LUN_number> VDISK=<storage_name> HOST=<host_name>
```

*LUN\_number* is 1 through 255.

### Switches

The following switches are available with the ADD LUN command.

#### **HOST=**

The name of the host to which the LUN is presented.

#### **VDISK=**

The name of the VDISK that is presented to the host.

### Required switches

Set the following switches so that Command View EVA recognizes the LUN you add with the ADD LUN command.

- HOST
- VDISK

### Examples

```
ADD LUN 12 VDISK="\Virtual Disks\act\payroll\ACTIVE"  
HOST=\Hosts\sanfran
```

```
ADD LUN 175 VDISK="\Virtual Disks\user_disk\ACTIVE"  
HOST=\Hosts\corporate
```

The first example adds LUN 12, which presents the ACTIVE virtual disk for the family *payroll* to the host called *sanfran*.

The second example adds LUN 175, which presents the ACTIVE virtual disk for the family *user\_disk* to the host called *corporate*.

## ADD SNAPSHOT

The ADD SNAPSHOT command creates a picture (not a full copy) of a virtual disk family's active virtual disk.

---

**Note:** Creating snapshots of virtual disks depends on your licensing level. Attempting to add a snapshot without the license returns an error message. This message indicates that you need to enter your licensing information in the Command View EVA. You must have a registered license.

---

### Syntax

```
ADD SNAPSHOT <snapshot_name> VDISK=<vdisk_to_snapshot>
```

### Switches

The following switches are available with the ADD SNAPSHOT command.

#### **ALLOCATION\_POLICY=**

Indicates how the space for the snapshot is allocated.

- **DEMAND**—Storage is allocated for the snapshot only when required. As the snapshot and the original storage's information diverges, the space allocated for the snapshot increases.
- **FULLY**—All storage required to fully contain a snapshot is allocated when the snapshot is created. This is the conservative method to create a snapshot, because adequate space is guaranteed for a full snapshot. This is the default.

#### **OS\_UNIT\_ID=**

The ID that is presented to the host operating system. If set to zero, no ID is presented to the host.

#### **REDUNDANCY=**

The amount of data protection set when creating the virtual disk. If not specified, the default is Vraid0.

- **Vraid0**—Provides no data protection. It distributes data among its member disks into stripes and uses all members to process I/O requests. This method has no overhead associated with duplication of information and provides the highest performance.
- **Vraid1**—Provides the highest level of data protection but uses the most space. It duplicates data written to one disk onto a backup disk. In a multidisk configuration, Vraid1 mirrors each pair of disks to each other. These disk pairs can then be striped to create a virtual disk.
- **Vraid5**—Provides a moderate level of data protection. This method distributes the parity information among all disk members. If one drive fails, the failed disk can be recreated after it is replaced.

**VDISK=**

The source virtual disk for this snapshot.

**WORLD\_WIDE\_LUN\_NAME=**

Sets the World Wide LUN Name.

**Required switch**

The following switch is required with the ADD SNAPSHOT command:

- **VDISK=**

**Examples**

```
ADD SNAPSHOT payroll_backup VDISK="\Virtual  
Disks\payroll\ACTIVE" ALLOCATION_POLICY=DEMAND  
  
ADD SNAPSHOT wed_nite_biz VDISK="\Virtual  
Disks\daily_biz\ACTIVE" ALLOCATION_POLICY=FULLY
```

The first example creates a snapshot *payroll\_backup* from the ACTIVE partition of Vdisk family *payroll* that uses capacity only as needed.

The second example creates a snapshot *wed\_night\_biz* from the ACTIVE partition of the family *daily\_biz* while reserving all capacity necessary to create the snapshot.

---

**Note:** To delete snapshots created by ADD SNAPSHOT, use the DELETE VDISK command.

---

## ADD SYSTEM

Use the ADD SYSTEM command to initialize an uninitialized storage system and to create a default disk group. You must select an uninitialized storage system before issuing the ADD SYSTEM command. If the system is already initialized, the command is rejected.

Use the SHOW SYSTEM command (see “[SHOW SYSTEM](#)” on page 100) to display the names of the available uninitialized storage systems. The uninitialized storage system display as *Uninitialized Storage System#*>, in which # represents a number. After initialization is complete, SSSU changes its default prompt back to *NoSystemSelected*>.

Because the system’s name changed from the uninitialized string to the name given with the ADD SYSTEM command, you must reselect the system. Issue the SELECT SYSTEM command using the new name of the system.

---

**Note:** [Appendix A](#) includes a configuration example that explains the sequence of commands you can use to initialize an uninitialized storage system (see “[Creating a simple configuration](#)” on page 106).

---

### Syntax

```
ADD SYSTEM <system_name>
```

### Switches

The following switches are available with the ADD SYSTEM command.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed, and the string must be enclosed in double quotes.

#### **CONSOLE\_LUN\_ID=**

The LUN used for console communication after system creation. If set to zero, no console LUN is presented to the host.

**DEVICE\_COUNT=**

The number of physical disks to use for the default disk group of this system. The limit is the number of available disks in the storage system. The default and minimum value is 8.

**DISKGROUP\_DISKTYPE=**

This parameter determines the types of disks to be considered for creating the disk group. The default value is ONLINE.

ONLINE: Online Fibre channel disks are considered for creating the default disk group.

NEAR-ONLINE: Near-Online Fibre channel disks are considered for creating the default disk group.

**SPARE\_POLICY=**

This parameter determines the amount, if any, of storage space set aside for safety in the event that disks fail. The default SPARE\_POLICY is SINGLE.

- NONE—Reserves no space within a disk group to allow for data reconstruction for failure of disk drives.
- SINGLE—Reserves space within a disk group to allow for data reconstruction for failure of a single disk drive. This is the default.
- DOUBLE—Reserves space within a disk group to allow for data reconstruction for failure of two disk drives.

---

**Note:** The space set aside is not in numbers of physical disks. It is the equivalent amount of storage space spread across all disks.

---

**Example**

```
ADD SYSTEM payroll DEVICE_COUNT=12 SPARE_POLICY=SINGLE
```

This example creates an initialized system *payroll* with 12 physical disks and a spare policy of *SINGLE*.

## ADD VDISK

Use the ADD VDISK command to create a virtual disk with a specified name and parameters.

---

**Note:** This command actually creates a virtual disk family and the ACTIVE partition. After you create the family, an initial virtual disk, "ACTIVE," is placed as the only virtual disk in the family. This is referred to as the active virtual disk. When you later refer to this virtual disk, specify \ACTIVE after the family name. See the examples that follow the descriptions of the switches.

---

### Syntax

```
ADD VDISK <storage_name> SIZE=<n>
```

The *n* is the virtual disk size in whole GBs.

### Switches

The following switches are available with the ADD VDISK command.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

#### **DISK\_GROUP=**

The disk group name for where you want to create the virtual disk. The disk group must already exist to use this switch. If not specified, the default disk group is used.

#### **MIRRORED\_WRITEBACK**

All writes are stored in mirrored caches. After the data is stored in both caches, the write is complete. If one mirrored cache is unusable, the write is not complete until the data is in both caches.



**NOMIRRORED\_WRITEBACK**

No writes are stored in mirrored caches. After the data is stored in one cache, the write is complete. The write is complete even if a mirror copy of the cache is not available.

**OS\_UNIT\_ID=**

The ID that is presented to the host operating system. If set to zero, no ID is presented to the host.

**PREFERRED\_PATH=**

The preferred controller path that is specified to handle all I/O for the virtual disks. If a controller fails, the path always reverts to the working controller.

- **PATH\_A\_BOTH**—Controller path A fails over to controller B. When controller A restarts, the virtual disks fail back to controller A. This is failover/failback mode.
- **PATH\_A\_FAILOVER**—Controller path A fails over to controller B. When controller A restarts, the virtual disks do not fail back over to controller A. This is failover-only mode.
- **PATH\_B\_BOTH**—Controller path B fails over to controller A. When controller B restarts, the virtual disks fail back to controller B. This is failover/failback mode.
- **PATH\_B\_FAILOVER**—Controller path B fails over to controller A. When controller B restarts, the virtual disks do not fail back over to controller B. This is failover-only mode.

**NOPREFERRED\_PATH**

Allows the I/O to be handled by either controller.

**READ\_CACHE**

Reads are satisfied from the controller's cache.

**NOREAD\_CACHE**

Reads are always satisfied from the physical disks, not the controller's cache.

**REDUNDANCY=**

The amount of data protection set when creating the virtual disk. If not specified, the default is Vraid0.

- **Vraid0**—Provides no data protection. It distributes data among its member disks into stripes and uses all members to process I/O requests. This method has no overhead associated with duplication of information and provides the highest performance.
- **Vraid1**—Provides the highest level of data protection but uses the most space. It duplicates data written to one disk onto a backup disk. In a multidisk configuration, Vraid1 mirrors each pair of disks to each other. These disk pairs can then be striped to create a virtual disk.
- **Vraid5**—Provides a moderate level of data protection. This method distributes the parity information among all disk members. If one drive fails, the failed disk can be recreated after it is replaced.

**SIZE=**

This is a required switch. Size of the storage to be created. You can specify size in whole gigabytes only, and fractions are not allowed (1 GB – 2000 GB, limited by the actual amount of space available within the disk group).

**WORLD\_WIDE\_LUN\_NAME=**

Sets the World Wide LUN Name.

---

**Note:** This switch is commonly used to allow a host to point to a new version of the virtual disk by giving the new virtual disk the same WWN as the old virtual disk.

---

**WRITE\_PROTECT**

Does not allow writing to the virtual disk for all presented LUNs/host.

**NOWRITE\_PROTECT**

Allows writing to the virtual disk for all presented LUNs/host.

**WAIT\_FOR\_COMPLETION**

Default. Waits until the command's operation completes before displaying the next SSSU prompt (and returning control) or before running another script line. For large (greater than 1 TB) virtual disks, using this option can result in a long period before the prompt displays.

## NOWAIT\_FOR\_COMPLETION

Returns control as soon as possible. Some operations are invalid until the initial operation completes in the background. For example, you cannot present the Vdisk (add a LUN) until the command completes.

### Required switch

The following switch is required with the ADD VDISK command:

- SIZE=

### Switch defaults

The following defaults are used for the ADD VDISK command when nothing is specified:

- MIRRORED\_WRITEBACK
- NOWRITE\_PROTECT
- OS\_UNIT\_ID= defaults to zero
- PREFERRED\_PATH= defaults to NOPREFERRED\_PATH
- REDUNDANCY= defaults to Vraid0
- DISK\_GROUP= “/Disk Groups/*Default Disk Group*”

---

**Note:** If the Default Disk Group has been renamed, not specifying a disk group will result in error.

---

- READ\_CACHE

### Examples

```
ADD VDISK "\Virtual Disk\scratch" SIZE=10 REDUNDANCY=VRAID5  
READ_CACHE
```

```
ADD VDISK "\Virtual Disks\engineering\gene_research" SIZE=2  
DISK_GROUP="\Disk Groups\small_disks" MIRRORED_WRITEBACK
```

The first example creates a 10-GB virtual disk *scratch* within the default disk group using read cache and a Vraid5 redundancy level.

The second example creates a 2-GB disk at Vraid0 redundancy level in the *small\_disks* group using mirrored writeback cache.

---

**Note:** Remember that this name is the <family\_name> and that “\Virtual Disks\<family\_name>\ACTIVE” is the name of the ACTIVE virtual disk. You must use this full path (enclosed in double quotes) when referring to it with any other commands.

---

## CAPTURE CONFIGURATION

### Syntax

```
CAPTURE CONFIGURATION <file_name>
```

CAPTURE CONFIGURATION queries the currently selected system and creates from one to five SSSU scripts. Not all systems require all five scripts. For example, step1A is always generated and may be the only file that is required. You use these scripts to recreate the original configuration and in some cases to assist in the recovery of any site.

The scripts output to the console unless you specify a file. The scripts generate five files. Given a file name, `_StepXX` is appended after the file name and before the extension. `XX` is the restore step name, which is 1A, 1B, 1C, 2, and 3. For example, specifying `CAPTURE CONFIGURATION newyear.txt` causes SSSU to create the files `newyear_Step1A.txt`, `newyear_Step1B.txt`, `newyear_Step1C.txt`, `newyear_Step2.txt`, and `newyear_Step3.txt`.

SSSU checks to see if any files by the names to be generated exist. If so, SSSU prompts you to replace existing files with the new files or abort the command completely.

When creating files that contain the scripts, a progress bar displays on the console. This command may take a long time to complete depending on the size of the configuration.

---

**Note:** Do not reconfigure the selected system while this command is executing.

---

### Step 1 scripts

The division of step 1 into three smaller steps facilitates data replication (DR) recovery.

#### Step 1A script

Creates the storage system, disk groups, hosts, and VDISKS that are not used for data replication, either source or destination, and LUNs for the created disks. Step1A creates a basic system that does not include DR VDISKS or groups. CAPTURE CONFIG always creates this step.

**Step 1B script**

Creates all source VDISKS used in DR groups on this controller.

**Step 1C script**

Presents all source VDISKS (creates LUNS) that are used for DR groups to their hosts. This step exists to assist in recovering from a DR failure in which a source site was lost. LUNs can be presented in their original configuration by running the correct step 1C script.

**Step 2 script**

Recreates all DR-specific configuration information for which this system is the source. This involves the configuration's source DR\_GROUPS and their members only. Presentations of remote VDISKS are not restored by this command (see step 3).

This step provides flexibility when a site is completely lost. You must run step 1A, step 1B, and step 1C on both source and destination systems before step 2 can be run.

---

**Note:** If no destination VDSIK is presented to a host, a blank script will be generated for script 3.

---

**Step 3 script**

Presents remote VDISKS (creates LUNS) used for DR groups to their hosts.

You must run step 2 on both source and destination systems before step 3 can be run.

**Restoring configurations**

To restore system configuration from captured scripts, you must run the following scripts in the following order on each system that is part of the DR configuration.

Before running a captured configuration script, you need to modify an existing script, run an additional script, or enter commands directly at the command line setup, as in steps 1 and 2.

Be aware that a specific captured configuration may not be recreated on some systems. For example, if a captured configuration demands 120 disks and the target system contains only 80, the captured configuration will not be successful.

1. Select a manager to specify which management appliance will perform the script commands.
2. Select the system Uninitialized Storage System# to specify the uninitialized storage system on which you want to recreate the captured configuration.
3. Run scripts 1A, 1B, and 1C successfully against ALL systems in the DR configuration.
4. Run the step 2 script successfully against ALL systems in the DR configuration.
5. Run the step 3 script successfully against ALL systems in the DR configuration.

Once these scripts have been run against all systems, the DR configuration is restored to its original state.

## Example

```
CAPTURE CONFIGURATION c:\sales.txt
```

This example creates the files *sales\_Step1A.txt*, *sales\_Step1B.txt*, *sales\_Step1C.txt*, *sales\_Step2.txt*, and *sales\_Step3.txt*.

---

**Note:** Refer to the [FILE](#) command on page 65 for further information about restoring configurations.

---

## CAPTURE VALIDATE

### Syntax

```
CAPTURE VALIDATE <file_name>
```

CAPTURE VALIDATE checks the specified script file for modification by calculating its checksum and comparing the result against the checksum saved in the file. A script may have been modified (failed the CAPTURE VALIDATE command) but may still work. SSSU runs a script even if its checksum shows file modification.

### Example

```
CAPTURE VALIDATE c:\reconfigure_sales_dr_step2.txt
```

This example determines whether the *reconfigure\_sales\_dr\_step2.txt* file was modified.

### Syntax

```
CAPTURE VALIDATE file_name
```

### User functionality

CAPTURE VALIDATE is a switch that determines if a captured script text file has been edited and modified after it was captured. This allows the user to determine whether the script is a perfect capture of the existing system or if someone has modified it after it was created.

Note that the specified script is not executed by this command; the script is checked only to see whether it's “original” or not.

When a script is captured by the CAPTURE commands, a header is written at the front of the text file and a checksum is written at the end. If CAPTURE VALIDATE does not see a correct header, it reports that the script specified is not a CAPTURE-generated script. If the checksum does not match, this command reports that the file has been modified.

If and only if a correct header is found and the checksum matches, this command reports that the CAPTURE script specified is unmodified.



## DELETE

Use the DELETE commands to remove systems, folders, disk groups, DR groups, VDISKS, hosts, and LUNs from the storage system. You can control the behavior of all DELETE commands with the SET OPTIONS command (see “[SET OPTIONS](#)” on page 83 for details on how to use the SAFE\_DELETE and NOSAFE\_DELETE switches).

---

**Note:** All of the DELETE commands require the full path from the root directory to the object you want to delete. Only the DELETE VDISK command has switches available.

---

## DELETE DR\_GROUP

Deletes the DR Group on the source and destination system. Be sure that VDISKS are unrepresented and that the source and destination are not communicating. The SET OPTION NOSAFE\_DELETE does not override this requirement. You can delete a DR Group on the source system at any time.

### Syntax

```
DELETE DR_GROUP <dr_group_name>
```

### Example

```
DELETE DR_GROUP "\Data Replication\payroll"
```

This example deletes the DR\_GROUP *payroll*.

## DELETE DISK\_GROUP

Use the DELETE DISK\_GROUP command to delete a disk group from the configuration. The command is rejected if any virtual disks are present in the disk group.

### Syntax

```
DELETE DISK_GROUP <group_name>
```

### Example

```
DELETE DISK_GROUP "\Disk Groups\human_resources"
```

This example deletes a disk group named *human\_resources*.

## DELETE FOLDER

Use the DELETE FOLDER command to delete the specified empty folder. You should ensure folders are empty before deleting them because the DELETE FOLDER command will be rejected otherwise. To delete multiple layers of folders, you must do so sequentially.

---

**Note:** Ensure folders are empty before deleting them because the DELETE FOLDER command is rejected if the folder has contents.

---

### Syntax

```
DELETE FOLDER <folder_name>
```

### Examples

```
DELETE FOLDER \Hosts\human_resources\
```

```
DELETE FOLDER \Hosts\colorado\colorado_springs\engineering\
```

The first example deletes a folder named *human\_resources* within the default Hosts folder.

The second example deletes the folder named *engineering* nested in the path *\Hosts\colorado\colorado\_springs*.

## DELETE HOST

### Syntax

```
DELETE HOST <host_name>
```

### Example

```
DELETE HOST \Hosts\development
```

This example deletes the host *development*.

## DELETE LUN

Use the DELETE LUN command to remove access to a virtual disk through the LUN from a host.

### Syntax

```
DELETE LUN <LUN_name>
```

### Example

```
DELETE LUN "\Hosts\accounting department\12"
```

This example removes access to LUN 12 from the host *accounting\_department*.

## DELETE SYSTEM

Use the DELETE SYSTEM command to permanently remove the system from an EVA. The system will no longer be accessible, and all data will be lost.



**Caution:** The DELETE SYSTEM command removes the entire system configuration. All customer data on the entire system is lost when you run this command. In addition, all information about LUNs and hosts on this system is lost.

---

The DELETE SYSTEM command will be rejected if VDISKS exist. Refer to the [SET](#) command for the NOSAFE\_DELETE behavior.

### Syntax

```
DELETE SYSTEM <system_name>
```

### Example

```
DELETE SYSTEM payroll
```

This example deletes the system *payroll*.

## DELETE VDISK

Use the DELETE VDISK command to permanently remove the specified virtual disk. The virtual disk is no longer accessible, and all data is lost.



**Caution:** The DELETE VDISK command removes the entire virtual disk from the storage system. All customer data on the storage specified is destroyed when you run this command. In addition, all information about the LUNs presented from this virtual disk is lost as well.

---

The DELETE VDISK command will be rejected if presented. See NO\_SAFE\_DELETE for behavior.

### Syntax

```
DELETE VDISK <storage_name>
```

### Switches

The following switches are available with the DELETE VDISK command.

#### WAIT\_FOR\_COMPLETION

Waits until the command's operation completes before displaying the next SSSU prompt (and returning control) or before running another script line. For large (greater than 1 TB) virtual disks, using this option can result in a long period before the prompt displays.

#### NOWAIT\_FOR\_COMPLETION

Default. Returns control as soon as possible. Some operations are invalid until the initial operation completes in the background.

### Examples

```
DELETE VDISK "\Virtual Disks\d12\ACTIVE" WAIT_FOR_COMPLETION
DELETE VDISK "\Virtual Disks\daily_biz\wed_nite_biz"
```

The first example deletes *ACTIVE* and the virtual disk family *d12*. *WAIT\_FOR\_COMPLETION* prevents the SSSU prompt from displaying until the virtual disk deletion is completed.



The second example deletes the snapshot *wed\_nite\_biz* and does not wait for completion.

---

**Note:** You cannot delete ACTIVE if a snapshot of the virtual disk exists. You must delete the snapshot and then delete ACTIVE.

---

## EMVERSION

The EMVERSION command is at the root level, as is help. This command provides the user with the element manager version and build number.

### Syntax

```
NoSystemSelected: emversion
```

### Switches

No additional keywords, parameters, or switches are available with the EMVERSION command.

### Example

```
Element manager information
Version: 3.10
Build: 14
```

---

**Note:** You will need to select a manager before using the EMVERSION command, but you do not need to select a storage system.

---

## EXERCISE\_DISK

This command allows users to test hard disks in the array for defects.

### Syntax

```
exercise_disk
```

### Switches

The following switches are available with the EXERCISE\_DISK command.

#### START

Begins testing, or exercising, the disks. The options are:

***All***

Tests all disks in a selected storage system.

***List of fully qualified disk names to exercise, comma separated and enclosed in quotes***

Tests specific disks as specified in the command.

#### STOP

Stops the testing.

#### SUMMARY

Provides a summary report of the testing. This can be used while testing is in progress to show status and also when testing is complete.

### Examples

```
exercise_disk start "\Disk Groups\Default Disk Group\Disk 002,  
\Disk Groups\Default Disk Group\Disk 004"  
exercise_disk start all
```

## EXIT

The EXIT command terminates the SSSU session.

If SSSU is accepting input from the terminal or the command line, EXIT causes the program to terminate.

If SSSU is processing a script from a file while encountering a FILE command, the behavior of the EXIT command depends on how the FILE command was issued:

- If the FILE command was issued from the command line, an EXIT command in the file causes SSSU to terminate.
- If the FILE command was issued from the terminal, an EXIT command in the file causes SSSU to return to accepting input from the terminal.
- If a file is currently executing as a result of another FILE command, an EXIT command in that file returns control to the previous file. SSSU does not run any commands in a script beyond the EXIT command. EXIT causes an immediate return to the “calling” layer.

### Syntax

```
EXIT
```

### Example

```
EXIT
```

SSSU terminates.

## FILE

The FILE command causes the current mode of input to suspend and redirects the scripting utility to accept input from the specified file. Either the end of the file or an EXIT command in the specified file causes SSSU to again accept input from the previous input source.

FILE commands can be nested. Nesting FILE commands means that a file being executed through a FILE command can have FILE commands within its command set. The only limitation on how deep FILE commands may be nested is based on the host system's resources.

### Syntax

```
FILE <filename>
```

An extension is not required by SSSU, although you can choose one appropriate to your environment.

### Examples

```
FILE snapd1.txt  
FILE d:\scripts\snapshots\d27.txt
```

The first example executes the file *snapd1.txt* from the current directory.

The second example executes a file *d27.txt* located in the specified directory.

## HELP

The HELP command displays information about using the SSSU help.

To get HELP on specific command syntax, enter a space and a question mark (?) wherever you would normally include a parameter, at any level of a command line. The help system will then list the options available for that parameter.

### Syntax

```
HELP
```

### Examples

```
HELP
```

```
Command syntax help is available at any point within a command  
by typing a '?'
```

The first example accesses HELP.

```
Add VDISK placeholder REDUNDANCY= ?
```

```
The system displays:
```

```
The options are:
```

```
VRAID0  
VRAID1  
VRAID5
```

The second example requests the display of the different valid values for the switch REDUNDANCY.

## REDISCOVER

This command tells Command View EVA to conduct rediscovery to find new storage systems or update the status of existing systems. For example, when communication has been lost and then restored, REDISCOVER could be used.

### Syntax

```
REDISCOVER
```

## RESTART

Use the RESTART command to restart a controller shown by the [SHOW POWER](#) command.

A power object is a controller.

### Syntax

```
RESTART <controller_name>
```

### Switches

The following switches are available for the RESTART command:

- ALL\_PEERS
- NOALL\_PEERS

These switches are described in the following paragraphs.

#### **ALL\_PEERS**

Restarts all peer controllers (both controllers) on this system.

#### **NOALL\_PEERS**

Default. Restarts only the specified controller.

### Examples

```
RESTART "\\Hardware\Rack 1\Enclosure 7\Controller B" ALL_PEERS  
RESTART "\\Hardware\Rack 1\Enclosure 7\Controller A" NOALL_PEERS
```

The first example restarts *Controller B*, including its peer controller.

The second example restarts *Controller A*, but not its peer controller.



**SELECT**

Use the SELECT command to select a system. After you select a system, the prompt changes to reflect the selected system.

## SELECT MANAGER

Use the SELECT MANAGER command to direct the command prompt to a selected Command View EVA or management appliance. All configuration commands run on the selected Command View EVA or management appliance and affect its storage system configuration. The *<manager\_name>* can be a server, appliance name, or an IP address. Unless the SAN administrator has changed the password, the default is “password.”

### Syntax

```
SELECT MANAGER <manager_name> USERNAME=<username>  
PASSWORD=<password>
```

### Switches

The switches available with the SELECT MANAGER command are described in the following paragraphs.

#### **PASSWORD=**

Password associated with the username that you use to access the Command View EVA.

#### **USERNAME=**

Username that you use to access the Command View EVA.

### Required switches

The following switches are required with the SELECT MANAGER command:

- **PASSWORD=**
- **USERNAME=**

### Example

```
SELECT MANAGER north_campus USERNAME=XXX PASSWORD=XXXX
```

This example selects the manager *north\_campus*. The username and password for the manager are validated in the background on every command.

## SELECT SYSTEM

Use the SELECT SYSTEM command to direct the command prompt to the selected system or manager. All configuration commands then affect the selected system. If the system name has spaces in it, the name must be enclosed in double quotes (“”).

### Syntax

```
SELECT SYSTEM <system_name>
```

### Examples

```
SELECT SYSTEM employees  
SELECT SYSTEM "payroll storage"
```

In the first example, the *employees* system is selected.

The system *payroll storage* is selected in the second example.

**SET**

Use the SET commands to change the properties of systems, disks, folders, DR groups, disk groups, hosts, and virtual disks. In addition, the SET commands let you reset existing storage settings and application options.

- All SET commands require the full path from the root directory to the object you want to rename.
- You can perform only one property change per SET command.

## SET DR\_GROUP

Use SET DR\_GROUP to set the mode of operation of an existing DR Group.

### Syntax

```
SET DR_GROUP <dr_group_name>
```

### Switches

The switches available with the SET DR\_GROUP command are described in the following paragraphs. Note that from the destination side, a user can set only the COMMENT and NAME switches.

#### **ACCESSMODE=**

The access rights for a connected host. Possible values are readonly and disable.

#### **ADD\_VDISK=**

Name of a source system VDISK to be added to the DR Group. A destination VDISK is automatically created on the destination system. When adding a VDISK to the DR Group, the following two optional switches may be specified:

##### **DESTINATION\_DISK\_GROUP=**

The disk group name on the destination system on which the VDISK is created. This switch can be specified only after the ADD\_VDISK switch.

##### **DESTINATION\_VDISK\_NAME=**

The name of the VDISK that is created on the destination system. The default name is the same as that of the source. This optional switch can be specified only after the ADD\_VDISK switch.

#### **COMMENT=**

An optional field for attaching comments to the DR Group being created. This string must be enclosed in quotes. The maximum number of characters for this switch is 64.

##### **DELETE\_VDISK=**

Name of a source system VDISK to be removed from the DR Group. The corresponding destination VDISK on the destination system is removed from the DR Group and deleted from the system.

**DETACH\_VDISK=**

Name of a source system VDISK to be removed from the DR Group. The corresponding destination VDISK on the destination system is removed from the DR Group but not deleted. The detached destination VDISK continues to exist as an independent VDISK.

**FAILOVER**

Reverses the roles of the DR Groups; the source becomes the destination and the destination becomes the source. This is used to both “fail over” to the destination site and to “fail back” when normal operation is to be resumed.

**FAILSAFE**

When the connection between the source and the destination system fails, all writes are immediately halted and none is reported as completing until either the connection is restored or NOFAILSAFE is set.

**NOFAILSAFE**

This is the default when you first create the DR Group. When the connection between the source and the destination system fails, writes are directed to the log until the connection is restored. At this point, the pending destination writes are pushed across to the destination system. If the log becomes full, the system makes a full copy of the source VDISK to the destination VDISK when the connection is restored. Otherwise, the log resynchs the source and destination VDISKs.

**NAME=**

This is the new name for the specified DR\_GROUP.

**SUSPEND**

Stops replication of data from the source to the destination. This causes the same behavior as if the connection failed.

**NOSUSPEND**

Allows replication of data from the source to the destination to resume. This causes the same behavior as if the connection was restored.

**WRITEMODE=**

Defines the I/O interaction between the destination and source side. Possible values are synchronous and asynchronous.

**Example**

```
SET DR_GROUP "\\Data Replication\Transactions" SUSPEND
```

This example suspends replication for the DR Group *Transactions*.

## SET DISK

The SET DISK command changes the properties of a disk.

### Syntax

```
SET DISK <name>
```

### Switches

The switches available for the SET DISK command are described in the following paragraphs.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

#### **NAME=**

New name to give to the specified disk.

### Example

```
SET DISK "\Disk Groups\Ungrouped Disks\Disk 005" NAME=5Disk
```

This example renames the disk *Disk 005* to *5Disk*.



## SET DISK\_GROUP

The SET DISK\_GROUP command changes the specified disk group properties.

### Syntax

```
SET DISK_GROUP <name>
```

### Switches

The switches available with the SET DISK\_GROUP command are described in the following paragraphs.

#### **ADD=**

The number of disk drives to add to the specified disk group.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the disk group. A maximum of 128 characters are allowed and the string must be enclosed in double quotes.

#### **DELETE=**

The name of the disk drive to be removed from the disk group. Data is not deleted when the disk is removed from the disk group.

#### **NAME=**

New name to give to the specified disk group.

#### **OCCUPANCY\_ALARM=**

The point when a defined percentage of space is used. When this point is reached, an event log is generated and sent to the Management Appliance (optionally the host system) informing the administrator that the group is becoming full. Do not use the percentage sign (%) after the number.

#### **SPARE\_POLICY=**

This parameter determines the amount, if any, of storage space set aside for use in the event that disks fail. The default is single.

- **NONE**—reserves no space within a disk group to allow for data reconstruction for failure of disk drives

- **SINGLE**—reserves space within a disk group to allow for data reconstruction for failure of a single disk drive
- **DOUBLE**—reserves space within a disk group to allow for data reconstruction for failure of two disk drives

---

**Note:** The space set aside is not in numbers of physical disks. It is the equivalent amount of storage space spread across all disks.

---

## Example

```
SET DISK_GROUP "\\Disk Groups\\pool" NAME="nuclear secrets"
```

This example renames the disk group *pool* with the new name *nuclear secrets*.

## SET FOLDER

The SET FOLDER command changes the specified folder properties.

### Syntax

```
SET FOLDER <name>
```

### Switches

The switches available for the SET FOLDER command are described in the following paragraphs.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed, and the string must be enclosed in double quotes.

#### **NAME=**

New name to give to the specified folder.

### Example

```
SET FOLDER "\Virtual Disks\top_secret\" NAME=everyone_knows
```

This example renames the folder *top\_secret* to *everyone\_knows*.

## SET HOST

The SET HOST command adds or deletes World Wide Names, sets the operating systems for a host, and adds comments to a specified host.

---

**Note:** Use the SET HOST command to add additional FCAs to hosts added to the storage system with the ADD HOST command.

---

### Syntax

```
SET HOST <name>
```

### Switches

The following switches are available with the SET HOST command.

#### **ADD\_WORLD\_WIDE\_NAME=**

Adds the World Wide Name of the host port.

#### **DELETE\_WORLD\_WIDE\_NAME=**

Deletes the World Wide Name of the host port.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed, and the string must be enclosed in double quotes.

#### **IP=**

This is the network IP address.

#### **NAME=**

New name given to the specified host.

#### **OPERATING\_SYSTEM=**

The type of operating system for the specified host. Choose one of the following operating systems, which are spelled here as SSSU expects:

- HPUX

- IBMAIX
- OPEN\_VMS
- SOLARIS
- TRU64
- WINDOWS
- CUSTOM = (You must include the equal sign after CUSTOM, and the value must be 16-digit hexadecimal characters.)

## Examples

```
SET HOST \Hosts\install OPERATING_SYSTEM=OPEN_VMS  
SET HOST \Hosts\install ADD_WORLD_WIDE_NAME=1000-0000-C922-36CA
```

The first example assigns OpenVMS as the operating system type for the host *install*.

The second example adds a WWN to the host *install*.

## SET MONITOR

The SET MONITOR command changes the specified monitor properties.

### Syntax

```
SET MONITOR <name>
```

### Switches

The following switches are available with the SET MONITOR command:

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the monitor. A maximum of 128 characters (including spaces) is allowed, and the string must be enclosed in double quotes.

#### **NAME=**

New name to give to the specified monitor.

### Example

```
SET MONITOR "\Hardware\Rack 1\Enclosure 7\Controller A"  
COMMENT="cabinet near the door"
```

This example adds the comment *cabinet near the door* to the monitor *Controller A*.

---

**Note:** Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

---

## SET OPTIONS

The SET OPTIONS command changes the SSSU characteristics.

**Note:** The options you set with the SET OPTIONS command are in effect for the current session only. Each time you start SSSU, the default options are reinstated.

### Syntax

```
SET OPTIONS
```

### Switches

The following switches are available with the SET OPTIONS command.

#### **COMMAND\_DELAY=**

The number of seconds (0–300 seconds) to wait between issuing commands when running a script using the FILE command. The default is 10 seconds. This switch has no effect when you are typing commands.

#### **NOCOMMAND\_DELAY**

Specifies no wait time between commands issued from within a FILE command.

#### **DISPLAY\_WIDTH=**

Sets the amount of characters displayed on a line for SHOW commands. The default is 80. If the SSSU output is to be parsed, it is useful to set a high line width. A long line keeps the lines from wrapping, making the output easier to cut, paste, and parse. Line widths from 70 to 500 may be specified.

#### **ON\_ERROR=**

- **CONTINUE**—In this mode, only the EXIT command causes the scripting utility to halt. This is the default and is recommended when you are manually typing commands.
- **EXIT\_ON\_ERROR**—Any error causes the scripting utility to exit with an error code. This mode is useful when requesting the entire script to halt immediately if errors occur while executing a script. This causes the scripting utility to halt on *any* kind of error: failed command, syntax error, or ambiguous command.

- **HALT\_ON\_ERROR**—This is similar to EXIT\_ON\_ERROR. Any error causes the scripting utility to halt but not exit until any key is pressed. Then the scripting utility exits with an error code. This allows you to see the error before the window closes on exit.

---

**Note:** When manually typing commands, HP does not recommend using the EXIT\_ON\_ERROR or HALT\_ON\_ERROR modes. You should avoid these modes because SSSU exits on *any* kind of error, including a typographical one.

---

### **RETRIES=**

Number of minutes to attempt “long” period retries, such as when the command view EVA service is busy or restarting. Specify between 1–10 minutes (inclusive). The default is four minutes.

### **NORETRIES**

Specify that you do not want the scripting utility to retry commands.

### **SAFE\_DELETE**

Specify if you want to delete all dependent or related objects before the specified object can be deleted. For example, if you specified a deletion of the VDisk that had LUNs presented, the DELETE VDISK command would be rejected with a message telling you that all the LUNs presented from this storage must be deleted before the storage can be deleted. This is the default.

You must always delete related objects if SAFE\_DELETE is specified.

### **NOSAFE\_DELETE**

Allows the deletion of an object even if related objects are present. The behavior of the NOSAFE\_DELETE command depends on the mode of operation (entering commands manually or running commands through a script) in which SSSU is running.

If you are entering commands at a terminal, the `Are you sure?` prompt displays and you must type `Yes` for the deletion to proceed. Note that you must type a capital `Y` followed by lowercase `es` for the deletion to proceed.

If the input is from a script (using the [FILE](#) command) or from the command line, the deletion proceeds without warning.





**Caution:** Using the NOSAFE\_DELETE switch can cause accidental deletion of VDISKS or presentations.

---

## Example

```
SET OPTION ON_ERROR=HALT_ON_ERROR NORETRIES
```

The example sets the options for SSSU to stop upon encountering an error and not to retry the commands.

## SET SYSTEM

Use the SET SYSTEM command to modify system properties.

### Syntax

```
SET SYSTEM <name>
```

### Switches

The following switches are available for the SET SYSTEM command.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed, and the string must be enclosed in double quotes.

#### **CONSOLE\_LUN\_ID=**

The LUN used for console communication. If set to zero, a console LUN is not presented to the host.

---

**Note:** This value is used for IBM AIX (set to zero), OpenVMS (required), and Tru64 UNIX (recommended). Other host operating systems ignore the value. See the host operating system installation guides for more information.

---

#### **NAME=**

New name to give to the specified system.

### Example

```
SET SYSTEM engineering NAME=accounting
```

This example renames the system *engineering* to *accounting*.

## SET VDISK

### Syntax

```
SET VDISK <storage_name>
```

### Switches

The following switches are available for the SET VDISK command.

#### **COMMENT=**

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

#### **MIRRORED\_WRITEBACK**

All writes are stored in mirrored caches. After the data is stored in both caches, the write is complete. If one mirrored cache is unusable, the write is not complete until the data is in both caches.

#### **NOMIRRORED\_WRITEBACK**

No writes are stored in mirrored caches. After the data is stored in one cache, the write is complete. The write is complete even if a mirror copy of a cache is not available.

#### **NAME=**

New name to give to the specified virtual disk.

#### **OS\_UNIT\_ID=**

The ID that is presented to the host operating system. If set to zero, an ID is not presented to the host.

---

**Note:** This is a required field for OpenVMS and is recommended for Tru64 UNIX. It is optional for all other operating systems.

---

#### **PREFERRED\_PATH=**

The preferred controller path that is specified to handle all I/O for the virtual disks. If a controller fails, the path always reverts to the working controller.

- **PATH\_A\_BOTH**—Controller path A fails over to controller B. When controller A restarts, the virtual disks fail back to controller A. This is failover/failback mode.
- **PATH\_A\_FAILOVER**—Controller path A fails over to controller B. When controller A restarts, the virtual disks do not fail back over to controller A. This is failover-only mode.
- **PATH\_B\_BOTH**—Controller path B fails over to controller A. When controller B restarts, the virtual disks fail back to controller B. This is failover/failback mode.
- **PATH\_B\_FAILOVER**—Controller path B fails over to controller A. When controller B restarts, the virtual disks do not fail back over to controller B. This is failover-only mode.

**NOPREFERRED\_PATH**

Allows the I/O to be handled by any controller.

**READ\_CACHE**

Reads are satisfied from the controller's cache.

**NOREAD\_CACHE**

Reads are always satisfied from media, not the controller's cache.

**SIZE=**

Indicates the new larger size (you cannot change to a size smaller than the original) of the storage in gigabytes. You can specify size in whole gigabytes only. Fractions are not allowed. The value of SIZE can be 2 GB through 2000 GB depending on the disk group's available space.

**WORLD\_WIDE\_LUN\_NAME=**

Sets the World Wide LUN Name on unpresented storage (either ACTIVE or snapshot). This switch is rejected when issued against storage that is presented to a host.

---

**Note:** This switch is commonly used to allow a host to point to a new version of a snapshot by giving the new snapshot the same WWN as the old snapshot.

---

**WRITE\_PROTECT**

Does not allow writing to the virtual disk for all presented LUNs.

**NOWRITE\_PROTECT**

Allows writing to the virtual disk for all presented LUNs.

**Example**

```
SET VDISK "\Virtual Disks\archive\ACTIVE" WORLD_WIDE_NAME_LUN_  
NAME=6000-1fe1-ff00-0000
```

The example assigns the World Wide LUN Name to the *ACTIVE* virtual disk of the *archive* family.

## SHOW

Use the SHOW commands to display information about various elements in the currently selected system.

---

**Note:** All of the SHOW commands require the full path from the root directory to the object you want to display. Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

---

The following switches are available for all SHOW commands.

- SHOW <element> FULL
- SHOW <element> NOFULL (This is the default.)

### Example

```
SHOW VDISK FULL
```

This example provides all information about all VDISKS.

### Example

```
SHOW VDISK NOFULL
```

This example lists all the VDISKS. Because NOFULL is the default, SHOW VDISK also list all VDISKS.

### Example

```
SHOW VDISK <name>
```

This example displays all the properties of a particular disk.

## SHOW DR\_GROUP

Displays a list of data replication groups for the currently selected system. If you specify the name of the DR Group, it displays detailed information about the specified DR Group.

### Syntax

```
SHOW DR_GROUP <dr_group_name>
```

### Example

```
SHOW DR_GROUP "\\Data Replication\\Transactions"
```

This example shows the DR\_Group *Transactions*.

## SHOW DISK

Use the SHOW DISK command to display disk configuration information for physical disks connected to the currently selected system.

### Syntax

```
SHOW DISK <disk_name>
```

### Examples

```
SHOW DISK  
SHOW DISK "\Disk Groups\Ungrouped Disks\Disk 039"
```

The first example displays a list of all the physical disks in the currently selected system.

The second example displays specific information about the physical “*Disk 039*.”



## SHOW DISK\_GROUP

The SHOW DISK\_GROUP command displays the disk groups currently configured by this system. If *<disk\_group\_name>* is specified, detailed information about the specified disk group is displayed.

### Syntax

```
SHOW DISK_GROUP <disk_group_name>
```

### Examples

```
SHOW DISK_GROUP  
SHOW DISK_GROUP "\Disk Groups\Default Disk Group"
```

The first example displays a list of the available disk groups.

The second example displays information about the default disk group.

## SHOW HOST

The SHOW HOST command displays the hosts currently configured on this system. If *<host\_name>* is specified, information about the specified host is displayed.

### Syntax

```
SHOW HOST <host_name>
```

### Examples

```
SHOW HOST  
SHOW HOST \Hosts\med_lab
```

The first example displays the list of available hosts on the currently selected system.

The second example displays specific information about the host *med\_lab*.

## SHOW LUN

The SHOW LUN command displays the LUNs currently configured on this system.

### Syntax

```
SHOW LUN <LUN_name>
```

### Examples

```
SHOW LUN  
SHOW LUN \Hosts\SVT165\13
```

The first example displays the list of available LUNs on the selected system.

The second example displays specific information about the LUN 13 on *\Hosts\SVT165*.

## SHOW MANAGER

The SHOW MANAGER command displays information about the currently selected manager.

### Syntax

```
SHOW MANAGER
```

### Switch

The switches FULL and NOFULL can be specified but are ignored. Specifying a specific manager name is also ignored. The SHOW MANAGER command always displays the information for the currently selected manager.

### Example

```
SHOW MANAGER
```

The example displays information about the currently selected manager.

## SHOW MONITOR

The SHOW MONITOR command displays the monitors currently installed on this system. If *<monitor\_name>* is specified, detailed information about that specified monitor is displayed.

### Syntax

```
SHOW MONITOR <monitor_name>
```

### Examples

```
SHOW MONITOR  
SHOW MONITOR "\Hardware\Rack 1\blue_cab"
```

The first example lists monitors available on the currently selected system.

The second example displays information about the monitor *blue\_cab*.

## SHOW OPTIONS

The SHOW OPTIONS command displays the current SSSU options.

### Syntax

```
SHOW OPTIONS
```

### Example

```
SHOW OPTIONS
```

The example shows the following current options set for SSSU:

```
Options:  
NOCOMMAND_DELAY  
DISPLAY_WIDTH = 80  
ON_ERROR = CONTINUE  
RETRIES = 4 (minutes)  
SAFE_DELETE
```

## SHOW POWER

The SHOW POWER command displays the controllers currently configured on this system. If you specify *<controller\_name>*, detailed information about that specific controller is displayed. Controllers are the only power objects.

### Syntax

```
SHOW POWER <controller_name>
```

### Examples

```
SHOW POWER  
SHOW POWER "\Hardware\Rack 1\Enclosure 7\Controller A"
```

The first example displays a list of all power objects within the selected system. The second example displays specific information about *Controller A*.

## SHOW SYSTEM

Use the SHOW SYSTEM command to display the systems currently managed by command view eva. If *<system\_name>* is specified, detailed information about that specified system is displayed.

### Syntax

```
SHOW SYSTEM <system_name>
```

### Examples

```
SHOW SYSTEM  
SHOW SYSTEM payroll
```

The first example displays a list of available systems on this manager.

The second example displays specific information about the system *payroll*.



## SHOW VDISK

The SHOW VDISK command displays the storage currently configured on this system. If *<storage\_name>* is specified, detailed information about the specified storage is displayed.

### Syntax

```
SHOW VDISK <virtualdiskname>
```

### Examples

```
SHOW VDISK  
SHOW VDISK "\Virtual Disks\scratch_disk\ACTIVE"
```

The first example displays a list of all virtual disks in the currently selected system.

The second example displays information about the specific active virtual disk *scratch\_disk\ACTIVE*.

## SHOW WORLD\_WIDE\_NAME

The SHOW WORLD\_WIDE\_NAME command displays the host World Wide Names (WWNs) that are visible to this system and not already assigned to a host.

### Syntax

```
SHOW WORLD_WIDE_NAME
```

### Example

```
SHOW WORLD_WIDE_NAME
```

This example displays all of the WWNs visible to this system that have not been assigned to a host.

## SHUTDOWN

Use the SHUTDOWN command to shut down any controller displayed by the [SHOW POWER](#) command.

### Syntax

```
SHUTDOWN <controller_name>
```

### Switches

The following switches are available with the SHUTDOWN command.

#### **ALL\_PEERS**

Shuts down all peer controllers (both controllers) on this system. This will also power off all drive enclosures.

#### **NOALL\_PEERS**

Default. Shuts down only the specified controller.

### Examples

```
SHUTDOWN "\Hardware\Rack 1\Enclosure 7\Controller A"  
NOALL_PEERS  
SHUTDOWN "\Hardware\Rack 1\Enclosure 7\Controller B" ALL_PEERS
```

The first example shuts down only *Controller A*.

The second example shuts down *Controller B* and the peer.



# Configuration Examples



This appendix provides examples that describe how to use commands to create a simple configuration and to view specific system information. The following topics are covered:

- [Creating a simple configuration](#), page 106
- [Using the SHOW commands](#), page 110

## Creating a simple configuration

This section presents a simple configuration example in two ways:

- Annotated with steps (page 106 through page 109)
- Sample output as it displays after executing all of the commands (page 109)

## Creating the storage system

- Select the manager with the `SELECT MANAGER` command.
  - View the systems available on the manager (Uninitialized Storage System1, if this is an uninitialized storage system) with the `SHOW SYSTEM` command.
  - Select the targeted cell for creation by issuing the command `SELECT SYSTEM Uninitialized Storage System1`.
  - Add (create) the storage system with the `ADD SYSTEM` command.
1. Start SSSU from anywhere on the same network as your management appliance.

The `NoSystemSelected>` prompt displays.

2. Enter the `SELECT MANAGER` command with the name of the manager and the logon information (a sample user name, password, and manager are used in the following example).

```
NoSystemSelected> SELECT MANAGER swma31k008 USERNAME=XXX  
PASSWORD=XXX  
NoSystemSelected>
```

---

**Note:** The password is set through the management appliance user interface.

---

3. At the prompt, enter the `SHOW SYSTEM` command to display all the available systems on this manager.

```
NoSystemSelected> SHOW SYSTEM  
Systems available on this Manager:  
Uninitialized Storage System1  
NoSystemSelected>
```

The display includes systems that are available on this manager. In the example, the default for an uninitialized storage system is one system named *Uninitialized Storage System1*.

4. Select the Uninitialized Storage System1 system with the SELECT SYSTEM command.

```
NoSystemSelected> SELECT SYSTEM "Uninitialized Storage System1"  
Uninitialized Storage System1>
```

The prompt changes to reflect the selected system (*Uninitialized Storage System1*).

5. Initialize the storage system with the ADD SYSTEM command, providing a name for the system (such as *Yekao*). This creates one system using 8 disks in the default Disk Group. The default number of disks is 8.

```
Uninitialized Storage System1> ADD SYSTEM Yekao  
NoSystemSelected>
```

---

**Note:** The prompt returns to `NoSystemSelected>` because the system's name has changed. You must reselect the system using the name you just gave it (`SELECT SYSTEM Yekao`).

---

## Creating and presenting a virtual disk

Here are the steps to follow to create a virtual disk and present the system to a host.

- Select the newly created system
  - Add a virtual disk (ADD VDISK) to the storage system
  - Add a host to the system (ADD HOST)
  - Add a LUN using a host and a VDISK
1. Use the SELECT SYSTEM command to select the system you just created.

```
NoSystemSelected> SELECT SYSTEM Yekao  
Yekao>
```

The prompt changes to reflect the object selected, in this case the system *Yekao*.

2. Create a virtual disk using the ADD VDISK command, supplying the virtual disk name and size in whole GBs.

```
Yekao> ADD VDISK Enicar SIZE=4  
Yekao>
```

This example creates a 4-GB virtual disk named *Enicar* with a Vraid0 redundancy.

---

**Note:** The virtual disk (storage) can be between 2 GB (minimum) and 2000 GBs (maximum). The increments must be in whole GBs.

---

To present this newly created virtual disk, you need to have at least one host within the storage system. This host must be added to the storage system configuration.

You can use the SHOW WORLD\_WIDE\_NAME command to see the available Fibre Channel Adapter World Wide Names.

3. Use the ADD HOST command to add a host to the storage system, supplying the World Wide Name (WWN) for the host port (FCA).

Use caution with WWN assignments as these are not verified against the existing adapter when assigned.

```
Yekao> ADD HOST \Hosts\SVT165  
WORLD_WIDE_NAME=1000-0000-c923-6735  
Yekao> SET HOST \Hosts\SVT165  
ADD_WORLD_WIDE_NAME=1000-0000-c923-6736
```



4. Add the LUN to the host by using the ADD LUN command, supplying the full path to the virtual disk and the host machine.

```
Yekao> ADD LUN 15 VDISK="\Virtual Disks\Enicar\ACTIVE"  
HOST=\Hosts\SVT165
```

Remember, for any path or name that contains spaces, you must enclose the entire string in double quotes, as in the example above.

Now LUN 15 is presenting the VDISK to the host.

### Sample Configuration Command Output

The following example shows the commands that were entered during the previous procedures (page 106 through page 109) without the interruption of the instructions.

```
NoSystemSelected> SELECT MANAGER swma31k008 USERNAME=XXX  
PASSWORD=XXX  
  
NoSystemSelected> SHOW SYSTEM  
  
Systems available on this Manager:  
  
Uninitialized Storage System1  
  
NoSystemSelected> SELECT SYSTEM "Uninitialized Storage System1"  
  
Uninitialized Storage System1> ADD SYSTEM Yekao  
  
NoSystemSelected> SELECT SYSTEM Yekao  
  
Yekao> ADD VDISK Enicar SIZE=4  
  
Yekao> ADD HOST SVT165 WORLD_WIDE_NAME=1000-0000-c923-6735  
  
Yekao> SET HOST \Hosts\SVT165 ADD_WORLD_WIDE_NAME=  
1000-0000-c923-6736  
  
Yekao> ADD LUN 13 VDISK="\Virtual Disks\Enicar\ACTIVE HOST=  
\Hosts\SVT165
```

## Using the SHOW commands

This section provides examples of using the SHOW command, based on the storage system, host, and virtual disk created in the previous section, “[Creating a simple configuration](#)” on page 106. The purpose of this presentation is to provide a sample of SSSU output.

To view information about various system attributes, you first select the system and then use the SHOW commands.

### SHOW SYSTEM

Use the SHOW SYSTEM command to view information about the selected system.

1. At the NoSystemSelected> prompt, enter SHOW SYSTEM.

```
NoSystemSelected> SHOW SYSTEM
Systems available on this Manager:
Yekao
Makawao
```

```
NoSystemSelected>
```

The available systems on this manager are displayed.

2. Select the system you want with the SELECT command.

```
NoSystemSelected>SELECT SYSTEM Yekao
```

3. Use the SHOW SYSTEM command to display information about the system.

```
Yekao> show system Yekao
Yekao information:
Initialized Enterprise System Properties:
      Name : Yekao
      Licenses : BASIC SNAPSHOT DRM
      Operational_State : Initialized - Good
      Total_storage_space : 947.78 GB
      Storage_space_used : 194.51 GB
      Available_storage_space : 753.26 GB
      Device_addition_policy : Manual
      Console_LUN_ID : 0
      Controller_time : 08-Jan-1947 15:01:00
      Node_world_wide_name : 5000-1FE1-5000-5C20
      Comments :
      ID : 08000710b4080560bc49010000d0000000000200
      ElementRefreshStatus : 0,0,0,0,0,0,0,0,3,0,0,0
```

```
Yekao>
```

## SHOW VDISK

Use the SHOW VDISK command to display information about the virtual disks (storage) within the selected system.

1. Enter the SHOW VDISK command (with a system selected) to display a list of available storage.

```
Yekao> show vdisk
Vdisks available on this System:
  \Virtual Disks\maui6228_1\ACTIVE
  \Virtual Disks\maui_1\ACTIVE
  \Virtual Disks\lanai_1\ACTIVE
Yekao>
```

---

**Note:** Remember that the ADD VDISK command creates a virtual disk family and you must include “\ACTIVE” when referring to the actual virtual disk.

---

2. Use the SHOW VDISK command to display information about the virtual disk.

```
Yekao> show vdisk "\Virtual Disks\maui6228_1\ACTIVE"
\Virtual Disks\maui6228_1\ACTIVE information:
Identification:
    Name : \Virtual Disks\maui6228_1\ACTIVE
    Family_Name : maui6228_1
    Hex_UUID : 6005-08b4-0001-49bc-0000-e000-000d-0000
    World_Wide_Lun_Name : 6005-08b4-0001-49bc-0000-e000-000d-0000
Attributes:
    Disk_Type : Original
    Disk_Group : \Disk Groups\Default Disk Group
    Capacity_Req : 25
    Capacity_Used : 25
    Sharing_Relationship : none
    Redundancy : vRaid5
    Write_Cache_Policy : Mirrored write-back
```

```

        Read_Cache_Policy : On
        Comm_Protocol :
        Write_Protect : No
        Enable_Disable : Enable
        OS_Unit_ID : 0
        Preferred_path : No preference
        Comments :
ID : 01200710b4080560bc49010000e0000000000d00
Condition_State:
Operational_State : 1
Date Time:
        Created : 28-Jun-1946 18:28:34
Presentation [0]:
        StorageClientName : \Hosts\maui_6228
        Lun : 1
DR_Attributes:
Remote_VDisk:
Yekao>

```

## SHOW HOST

Use the SHOW HOST command to display information about the hosts available on the selected storage system.

1. With the system selected, enter SHOW HOST to display the list of available hosts.

```

Yekao> show host

Hosts available on this System:

\Hosts\maui
\Hosts\maui_6228
\Hosts\lanai

Yekao>

```

2. Use the SHOW HOST command, followed by the specific host name, to display information about the host.

---

**Note:** Remember that you need to type the full path to the host object and that if there are any spaces within the path or host name, the entire string needs to be enclosed within double quotes.

---

```
Yekao> show host "\Hosts\maui"
\Hosts\maui information:
  Identification:
    Name : \Hosts\maui
    IPAddress : Dynamic IP Assignment
    Status : Initialized - Good
    Operating_System : IBMAIX
  Fibre_Channel_adapter_ports:
  Fibre_Channel_adapter_ports [0]:
    Port : 5005-02e0-1000-09fe
  Fibre_Channel_adapter_ports [1]:
    Port : 5005-02e0-1000-018b
  Presentation:
  Presentation [0]:
    Lun : 1
    Virtual_disk : \Virtual Disks\maui_1\ACTIVE
    Comments :
    ID : 00800710b4080560bc49010000d0000000001b00
Yekao>
```

## SHOW LUN

Use the SHOW LUN command to display information about specific LUNs.

1. With the system selected, enter the SHOW LUN command to display the list of available LUNs.

```
Yekao> show lun
LUNs available on this Cell:
  \Hosts\maui_6228\1
  \Hosts\maui\1
  \Hosts\lanai\1
```

2. Use the SHOW LUN command, with the specific LUN named, to display the specific LUN information.

```
Yekao> show lun "\Hosts\maui\1"
\Hosts\maui\1 information:
      Name : \Hosts\maui\1
      StorageName : \Virtual Disks\maui_1\ACTIVE
      StorageclientName : \Hosts\maui
      ID : 00880710b4080560bc49010000e0000000000b00
Yekao>
```



## glossary

This glossary defines terms used in this guide or related to this product and is not a comprehensive glossary of computer terms.

### **active virtual disk**

A virtual disk is a simulated disk drive created by the controllers as storage for one or more hosts. An active virtual disk is accessible by one or more hosts for normal storage. An active virtual disk and its snapshot, if one exists, constitute a virtual disk family. An active virtual disk is the only necessary member of a virtual disk family.

*See also* **virtual disk**, **virtual disk copy**, **virtual disk family**, and **snapshot**.

### **array**

All the physical disk drives in a storage system that are known to and under the control of a controller pair.

### **array controller**

*See* **controller**.

### **block**

*Also called a sector.* The smallest collection of consecutive bytes addressable on a disk drive. In integrated storage elements, a block contains 512 bytes of data, error codes, flags, and the block address header.

### **cache**

High-speed memory that sets aside data as an intermediate data buffer between a host and the storage media. The purpose of cache is to improve performance.

*See also* **read cache**, **write cache**, and **mirrored cache**.

### **command view EVA**

A graphical user interface through which users control and monitor a storage system. The software can be installed on one or more management appliances in a fabric. Each Command View EVA installation is a management agent.

### **communication logical unit number (LUN)**

*See* **console LUN**.

**console LUN**

A SCSI-3 virtual object that makes a controller pair accessible by the host before any virtual disks are created. *Also called a **communication LUN**.*

**console LUN ID**

The ID that can be assigned when a host operating system requires a unique ID. The console LUN ID is assigned by the user, usually when the storage system is initialized.

*See also **console LUN**.*

**controller**

A hardware/firmware device that manages communications between host systems and other devices. Controllers typically differ by the type of interface to the host and provide functions beyond those the devices support.

**controller pair**

Two interconnected controller modules that together control a physical disk array. A controller pair and the disk array together constitute a storage system.

**default disk group**

The first disk group created at the time the system is initialized. The default disk group can contain the entire set of physical disks in the array or just a few of the disks.

*See also **disk group**.*

**disk**

A physical storage device (disk drive) connected to the system.

**disk failure protection**

A method by which a controller pair reserves drive capacity to take over the functionality of a failed or failing physical disk.

**DISK\_GROUP**

A collection of disks from which storage is created.

**DR\_GROUP**

One or more data replication groups that preserve write ordering and share a common log.

**EMU**

Environmental Monitoring Unit. An element that monitors the status of an enclosure, including the power, air temperature, and blower status. The EMU detects problems and displays and reports these conditions to a user and the controller. In some cases, the EMU implements corrective action.



**Enterprise Virtual Array**

The hp name used to describe the storage system that includes controllers, storage devices, enclosures, cables, and power supplies. Also called the **Enterprise Storage System**.

**environmental monitoring unit**

*See* EMU.

**event log**

System information sent to the Management Appliance or host operating system.

**fabric**

A Fibre Channel fabric switch or two or more interconnected Fibre Channel switches allowing data transmission.

**fabric port**

A port that is capable of supporting an attached arbitrated loop. This port on a loop has the AL\_PA hexadecimal address 00 (loop ID 7E), giving the fabric the highest priority access to the loop. A loop port is the gateway to the fabric for the node ports on a loop.

**failover**

The process that takes place when one controller assumes the workload of a failed companion controller. Failover continues until the failed controller is operational.

**FC HBA**

Fibre Channel Host Bus Adapter. An interchangeable term for Fibre Channel adapter.

*See also* FCA.

**FCA**

Fiber Channel Adapter. An adapter used to connect the host server to the fabric. *Also* called a Host Bus Adapter (HBA) or a Fibre Channel Host Bus Adapter (FC HBA).

*See also* FC HBA

**fiber**

The optical media used to implement Fibre Channel.

**fibre**

The international spelling that refers to the Fibre Channel standards for optical media.

**Fibre Channel**

A data transfer architecture designed for mass storage devices and other peripheral devices that require very high bandwidth.

**Fibre Channel adapter**

*See* FCA.

## **GB**

Gigabyte. A unit of measurement defining either:

- A data transfer rate.

*See also* **GBps**

- A storage or memory capacity of 1,073,741,824 ( $2^{30}$ ) bytes.

## **GBps**

Gigabytes per second. A measurement of the rate at which the transfer of bytes of data occurs. A GBps is a transfer rate of 1,000,000,000 ( $10^9$ ) bits per second.

*See also* **GB**.

## **Giga (G)**

The notation to represent  $10^9$  or 1 billion (1,000,000,000).

## **HBA**

Host Bus Adapter.

*See* **FCA**.

## **host**

A system that has data path access to the virtual disk through a logical unit number (LUN) defined on the system.

## **Host Bus Adapter**

*See* **FCA**.

## **host computer**

*See* **host**.

## **host ports**

A connection point to one or more hosts through a Fibre Channel fabric. A host is a computer that runs user applications and that uses (or can potentially use) one or more of the virtual disks that are created and presented by the controller pair.

## **host-side ports**

*See* **host ports**.

## **hp StorageWorks**

The hp trademarked name used to describe the set of rack-mounted enclosures containing controllers, transceivers, I/O modules, EMUs, disk drives, cables, blowers, and power supplies used to design and configure a solution-specific storage system.

**initialization**

A process that prepares a storage system for use. Specifically, the system binds controllers together as an operational pair and establishes preliminary data structures on the disk array. Initialization also sets up the first disk group, called the default disk group.

**LUN**

Logical Unit Number. The host-accessible presentation of a virtual disk on the system.

**manager**

The object that controls configuration of the system. The StorageWorks Command View EVA is the manager in the case of the Enterprise Virtual Array.

**MB**

Megabyte. A term defining either:

- A data transfer rate.  
*See also MBps.*
- A measure of either storage or memory capacity of 1,048,576 ( $2^{20}$ ) bytes.

**MBps**

Megabytes per second. A measure of bandwidth or data transfers occurring at a rate of 1,000,000 ( $10^6$ ) bytes per second.

**Mega**

A notation denoting a multiplier of 1 million (1,000,000).

**mirrored caching**

A process in which half of each controller's write cache mirrors the companion controller's write cache.

**mirroring**

The act of creating an exact copy or image of data.

**monitor**

An element that monitors the status of an enclosure, including the power, air temperature, and blower status.

**online/near-online drive**

An online drive is a normal, high-performance drive, while a near-online drive is a lower-performance drive.

## **OSM**

Open SAN Manager. A centralized, appliance-based monitoring and management interface that supports multiple applications, operating systems, hardware platforms, storage systems, tape libraries, and SAN-related interconnect devices. It is included with and resides on the Management Appliance, a single aggregation point for data management.

## **parity**

A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even. Parity information can be used to correct corrupted data.

## **parity bit**

A binary digit added to a group of bits that checks to see if there are errors in the transmission.

## **parity check**

A method of detecting errors when data is sent over a communications line. With even parity, the number of ONE bits in a set of binary data should be even. With odd parity, the number of ONE bits should be odd.

## **password**

A security interlock whose purpose is to allow:

- A management agent control of only certain storage systems
- Only certain management agents control of a storage system

## **physical disk**

A disk drive mounted in a disk drive enclosure that communicates with a controller pair through the device-side Fibre Channel loops. A physical disk is hardware with embedded software, as opposed to a virtual disk, which is constructed by the controllers. Only the controllers can communicate directly with the physical disks.

The physical disks, in aggregate, are called the array and constitute the storage pool from which the controllers create virtual disks.

## **physical disk array**

*See* **array**.

## **port**

A Fibre Channel connector on a Fibre Channel device.

## **port\_name**

A 64-bit unique identifier assigned to each Fibre Channel port. The port\_name is communicated during the login and port discovery processes.

**preferred path**

A preference for which controller of the controller pair manages the virtual disk. This preference is set by the user through the Command View EVA when creating the virtual disk. A user can change the preferred path of a virtual disk at any time. The main purpose of selecting a preferred path is load balancing.

**read ahead caching**

A cache management method used to decrease the subsystem response time to a read request by allowing the controller to satisfy the request from the cache memory rather than from the disk drives.

**read caching**

A cache method used to decrease subsystem response times to a read request by allowing the controller to satisfy the request from the cache memory rather than from the disk drives. Reading data from cache memory is faster than reading data from a disk. The read cache is specified as either on or off for each virtual disk. The default state is on.

**redundancy**

- Element Redundancy—The degree to which logical or physical elements are protected by having another element that can take over in case of failure. For example, each loop of a device-side loop pair normally works independently but can take over for the other in case of failure.
- Data Redundancy—The level to which user data is protected. Redundancy is directly proportional to cost in terms of storage usage; the greater the level of data protection, the more storage space is required.

**SCSI (small computer system interface)**

- An American National Standards Institute (ANSI) interface that defines the physical and electrical parameters of a parallel I/O bus used to connect computers and a maximum of 16 bus elements.
- The communication protocol used between a controller pair and the hosts. Specifically, the protocol is FC-AL or SCSI on a Fibre Channel. SCSI is the higher command-level protocol and Fibre Channel is the low-level transmission protocol. The controllers have full support for SCSI-2; additionally, they support some elements of SCSI-3.

**SCSI-3**

The ANSI standard that defines the operation and function of Fibre Channel systems.

**snapshot**

A temporary virtual disk that reflects the contents of another virtual disk at a particular point in time. A snapshot operation is done only on an active virtual disk. The active disk and its snapshot constitute a virtual disk family.

*See also* **active virtual disk**, **virtual disk copy**, and **virtual disk family**.

**spare policy**

This is the disk failure protection level set for the system and disk groups.

*See also* **disk failure protection**.

**SSN**

Storage System Name. A user-defined, unique 20-character name that identifies a specific storage system.

**storage pool**

The aggregated blocks of available storage in the total physical disk array.

**storage system**

The controllers, storage devices, enclosures, cables, and power supplies and their software.

**Storage System Name**

*See* SSN.

**Storage System Scripting Utility (SSSU)**

A command-line application supplied in the host operating system kits that lets you configure and control HSV controllers.

**switch**

An electro-mechanical device that initiates an action or completes a circuit.

**system**

A controller or controller pair that is treated conceptually as one controller. In the Command View EVA, the system is the name you give the initialized storage system.

**TB**

Terabyte. A term defining either:

- A data transfer rate.

*See also* **TBps**.

- A measure of either storage or memory capacity of 1,099,511,627,776 ( $2^{40}$ ) bytes.

**TBps**

Terabytes per second. A data transfer rate of 1,000,000,000,000 ( $10^{12}$ ) bytes per second.

**uninitialized storage system**

A state in which the storage system is not ready for use.

*See also* **initialization**.

**VDisk (virtual disk)**

A collection of blocks created on one or more disks that can be used by a host for information storage and retrieval.

The host computer sees the virtual disk as “real,” with the characteristics of an identical physical disk.

*See also* **active virtual disk**, **virtual disk copy**, **virtual disk family**, and **virtual disk snapshot**.

**virtual disk copy**

A clone or exact replica of another virtual disk at a particular point in time. Only an active virtual disk can be copied. A copy immediately becomes the active disk of its own virtual disk family.

*See also* **active virtual disk**, **virtual disk family**, and **virtual disk snapshot**.

**virtual disk family**

A virtual disk and its snapshot, if a snapshot exists, constitute a family. The original virtual disk is called the active disk. When you first create a virtual disk family, the only member is the active disk.

*See also* **active virtual disk**, **virtual disk copy**, and **virtual disk snapshot**.

**virtual disk snapshot**

*See* **snapshot**.

**Vraid0**

A virtualization technique that provides no data protection. Data host is broken down into chunks and distributed on the disks comprising the disk group from which the virtual disk was created. Reading and writing to a Vraid0 virtual disk is very fast and makes the fullest use of the available storage, but there is no data protection (redundancy) unless there is parity.

**Vraid1**

A virtualization technique that provides the highest level of data protection. All data blocks are mirrored or written twice on separate physical disks. For read requests, the block can be read from either disk, which can increase performance. Mirroring takes the most storage space because twice the storage capacity must be allocated for a given amount of data.

## **Vraid5**

A virtualization technique that provides moderate data protection using parity striping. Parity is a data protection mechanism for a striped virtual disk. A striped virtual disk is one whose data to and from the host is broken down into chunks and distributed on the physical disks comprising the disk group in which the virtual disk was created. If the striped virtual disk has parity, another chunk (a parity chunk) is calculated from the set of data chunks and written to the physical disks. If one of the data chunks becomes corrupted, the data can be reconstructed from the parity chunk and the remaining data chunks.

## **World Wide Name**

*See* **WWN**.

## **write back caching**

A controller process that notifies the host that the write operation is complete when the data is written to the cache. This occurs before transferring the data to the disk. Write back caching improves response time since the write operation completes as soon as the data reaches the cache. As soon as possible after caching the data, the controller then writes the data to the disk drives.

## **write caching**

A process when the host sends a write request to the controller and the controller places the data in the controller cache module. As soon as possible, the controller transfers the data to the physical disk drives.

## **WWN**

World Wide Name. A unique Fibre Channel identifier consisting of a 16-character hexadecimal number. A WWN is required for each Fibre Channel communication port.



**A****ADD COPY**

examples 31

switches

DISK\_GROUP= 30

OS\_UNIT\_ID= 30, 40

VDISK= 31

WORLD\_WIDE\_LUN\_NAME= 31

**ADD DISK\_GROUP**

description 34

example 35

switches

DEVICE\_COUNT= 34

OCCUPANCY\_ALARM= 35

SPARE\_POLICY= 35

**ADD DR\_GROUP**

description 32

example 33

Syntax 32

**ADD FOLDER**

description 36

examples 36

switch, COMMENT= 36

syntax 36

**ADD HOST**

description 37

example 38

switches

COMMENT= 37

OPERATING\_SYSTEM= 37

WORLD\_WIDE\_NAME= 38

syntax 37

using 108

**ADD LUN**

description 39

examples 39

switches

HOST= 39

VDISK= 39

syntax 39

using 109

**ADD SNAPSHOT**

description 40

examples 41

switches

ALLOCATION\_POLICY= 40

VDISK= 41

WORLD\_WIDE\_LUN\_NAME= 41

syntax 40

**ADD SYSTEM**

description 42

example 43

switches

COMMENT= 34, 36, 37, 42, 44, 86

CONSOLE\_LUN\_ID= 42

DEVICE\_COUNT= 43

SPARE\_POLICY= 43

syntax 42

using 107

**ADD VDISK**

description 44

examples 47

switch defaults 47

switches

DISK\_GROUP= 44

- MIRRORED\_WRITEBACK [44](#)
- NOPREFERRED\_PATH [45](#)
- NOREAD\_CACHE [45](#)
- NOWRITE\_PROTECT [46](#)
- OS\_UNIT\_ID= [45](#)
- PREFERRED\_PATH= [45](#), [87](#)
- READ\_CACHE [45](#)
- REDUNDANCY= [30](#), [40](#), [45](#)
- SIZE= [46](#)
- WORLD\_WIDE\_LUN\_NAME= [46](#)
- WRITE\_PROTECT [46](#)
- syntax [44](#)
- using [108](#)
- audience [14](#)
- authorized reseller, HP [12](#)

## C

- CAPTURE CONFIGURATION
  - description [49](#)
  - example [51](#)
  - restoring configurations [50](#)
  - syntax [49](#)
- CAPTURE VALIDATE
  - description [52](#)
  - example [52](#)
  - syntax [52](#)
- configuration
  - creating [106](#)
  - importance of capturing [21](#)
- CONSOLE\_LUN\_ID
  - and operating systems [42](#)
- conventions
  - document [10](#)
  - text symbols [10](#)
- creating
  - and presenting a virtual disk [108](#)
  - simple configuration [106](#)
  - the storage system [106](#)

## D

- DELETE commands [53](#)

- DELETE DISK\_GROUP
  - description [55](#)
  - example [55](#)
  - syntax [55](#)
- DELETE DR\_GROUP
  - example [54](#)
  - syntax [54](#)
- DELETE FOLDER
  - description [56](#)
  - syntax [56](#)
- DELETE HOST
  - syntax [57](#)
- DELETE LUN
  - description [58](#)
  - syntax [58](#)
- DELETE SYSTEM
  - description [59](#)
  - syntax [59](#)
- DELETE VDISK
  - description [60](#)
  - examples [60](#)
  - syntax [60](#)
- DISK
  - SET [76](#)
  - SHOW [92](#)
- DISK\_GROUP
  - ADD [34](#)
  - DELETE [55](#)
  - SET [77](#)
  - SHOW [93](#)
- document
  - conventions [10](#)

## E

- Enterprise Virtual Array, and SSSU [14](#)
- EXIT
  - description [64](#)
  - example [64](#)
  - syntax [64](#)

**F**

## FILE

- description 65
- examples 65
- syntax 65

## FOLDER

- ADD 36
- DELETE 56
- SET 77

formatting and output of returned data 14

**G**

getting help 12

**H**

## HELP

- description 66
- example 66
- syntax 66, 67

help, obtaining 12

## HOST

- ADD 37
- SET 80
- SHOW 94

## HP

- authorized reseller 12
- storage website 12
- technical support 12

HP-UX 37, 80

HSV controllers 14

**I**

IBM AIX 37, 81

- and the CONSOLE\_LUN\_ID 86

installing SSSU 15

**L**

## LUN

- ADD 39
- DELETE 58
- SHOW 95

**M**

## MANAGER

- SHOW 96

## MONITOR

- SET 82
- SHOW 97

**N**

## naming

- disk group example 23
- host example 23
- paths and 23
- virtual disk example 23

**O**

OpenVMS 37, 81

- and the CONSOLE\_LUN\_ID 86

## operating systems

- how named in SSSU 37
- setting with ADD HOST 80

## OPTIONS

- SET 83
- SHOW 98

options, setting for SSSU 83

output of returned data, formatting 14

overview, of SSSU 14

**P**

paths and naming 23

POWER, SHOW 99

presenting a virtual disk, creating and 108

**R**

RESTART 67, 68

- description 67, 68

examples 68

## switches

- ALL\_PEERS 68, 103
- NOALL\_PEERS 68, 103

syntax 68

restoring configurations 50

returned data, formatting and output of [14](#)

## S

SELECT [69](#)

SELECT MANAGER

examples [70](#)

switches

PASSWORD= [70](#)

USER= [70](#)

syntax [70](#)

SELECT SYSTEM

examples [71](#)

syntax [71](#)

using [107](#), [108](#)

SET DISK

description [76](#)

example [76](#)

switches

COMMENT= [76](#)

NAME= [76](#)

syntax [76](#)

SET DISK\_GROUP

description [77](#)

example [78](#)

switches

ADD= [77](#)

COMMENT= [77](#)

DELETE= [77](#)

NAME= [77](#)

OCCUPANCY\_ALARM= [77](#)

syntax [77](#)

SET DR\_GROUP

description [73](#)

example [75](#)

syntax [73](#)

SET FOLDER

description [77](#)

example [79](#)

switches

COMMENT= [79](#)

NAME= [79](#)

syntax [79](#)

SET HOST

description [80](#)

examples [81](#)

switches

ADD\_WORLD\_WIDE\_NAME= [80](#)

COMMENT= [80](#)

DELETE\_WORLD\_WIDE\_NAME= [80](#)

OPERATING\_SYSTEM= [80](#)

syntax [80](#)

SET MONITOR

description [82](#)

example [82](#)

switches

COMMENT= [82](#)

NAME= [82](#)

syntax [82](#)

SET OPTIONS

description [83](#)

examples [85](#)

switches

COMMAND\_DELAY= [83](#)

DISPLAY\_WIDTH= [83](#)

NOCOMMAND\_DELAY [83](#)

NORETRIES= [84](#)

NOSAFE\_DELETE [84](#)

ON\_ERROR= [83](#)

RETRIES= [84](#)

SAFE\_DELETE [84](#)

syntax [83](#)

SET SYSTEM

description [86](#)

example [86](#)

switches

CONSOLE\_LUN\_ID= [86](#)

NAME= [86](#)

syntax [86](#)

SET VDISK

description [87](#)

examples [89](#)

switches

MIRRORED\_WRITEBACK [44](#), [87](#)

NAME= [87](#)

- NOMIRRORED\_WRITEBACK [45](#), [87](#)
- NOPREFERRED\_PATH [88](#)
- NOREAD\_CACHE [88](#)
- NOWRITE\_PROTECT [89](#)
- OS\_UNIT\_ID= [87](#)
- READ\_CACHE [88](#)
- SIZE= [88](#)
- WORLD\_WIDE\_LUN\_NAME= [31](#), [88](#)
- WRITE\_PROTECT [89](#)
- syntax [87](#)
- SHOW
  - commands [90](#)
  - output samples [110](#)
- SHOW DISK
  - description [92](#)
  - examples [92](#)
  - syntax [92](#)
- SHOW DISK\_GROUP
  - description [93](#)
  - examples [93](#)
  - syntax [93](#)
- SHOW DR\_GROUP
  - description [91](#)
  - example [91](#)
  - syntax [91](#)
- SHOW HOST
  - description [94](#)
  - examples [94](#)
  - output samples [112](#)
  - syntax [94](#)
- SHOW LUN
  - description [95](#)
  - examples [95](#)
  - output samples [113](#)
  - syntax [95](#)
- SHOW MANAGER
  - description [96](#)
  - examples [96](#)
  - syntax [96](#)
- SHOW MONITOR
  - description [97](#)
  - examples [97](#)
  - syntax [97](#)
- SHOW OPTIONS
  - description [98](#)
  - example [98](#)
  - syntax [98](#)
- SHOW POWER
  - description [99](#)
  - examples [99](#)
  - syntax [99](#)
- SHOW SYSTEM
  - description [100](#)
  - examples [100](#)
  - output samples [110](#)
  - syntax [100](#)
  - using [106](#)
- SHOW VDISK
  - description [101](#)
  - examples [101](#)
  - output samples [111](#)
  - syntax [101](#)
- SHOW WORLD\_WIDE\_NAME
  - description [102](#)
  - example [102](#)
  - syntax [102](#)
- SHUTDOWN
  - description [103](#)
  - examples [103](#)
  - syntax [103](#)
- simple configuration, creating a [106](#)
- SNAPSHOT, ADD [40](#)
- SSSU
  - installing [15](#)
  - naming operating systems [37](#)
  - overview [14](#)
  - setting interface options [83](#)
- storage system, creating [106](#)
- Sun Solaris [37](#), [81](#)
- switch defaults, ADD VDISK [47](#)
- symbols in text [10](#)
- SYSTEM
  - ADD [42](#)
  - DELETE [59](#)

SET [86](#)

SHOW [100](#)

## T

technical support, HP [12](#)

text symbols [10](#)

Tru64 UNIX [37](#), [81](#)

and the `CONSOLE_LUN_ID` [86](#)

## V

VDISK

ADD [44](#)

DELETE [60](#)

SET [87](#)

SHOW [101](#)

virtual disk, creating and presenting a [108](#)

Vraid0 [30](#), [41](#), [46](#)

Vraid1 [30](#), [34](#), [41](#), [46](#)

Vraid5 [30](#), [34](#), [41](#), [46](#)

## W

websites, HP storage [12](#)

Windows 2000 [38](#), [81](#)

Windows NT [38](#), [81](#)

`WORLD_WIDE_LUN_NAME`, use with  
snapshots [31](#)

`WORLD_WIDE_NAME`, SHOW [102](#)